Handoff Management for Infotainment Services over Vehicular Networks

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Author: Sergi Reñé
Advisors: Dr. Óscar Esparza and Dr. Juanjo Alins

A dissertation submitted to the Department of Network Engineering and the committee on graduate studies of Universitat Politècnica de Catalunya in partial fulfillment of the requirements for the degree of *Doctor of Philosophy*

May 2015

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola de Doctorat

Acta de qualificació de tesi doctoral

Curs acadèmic:

Nom i cognoms

Programa de doctorat

Unitat estructural responsable del programa

Resolució del Tribunal

Reunit el Tribunal designat a l'efecte, el doctorand / la doctoranda exposa el tema de la seva tesi doctoral titulada

____

____.

Acabada la lectura i després de donar resposta a les qüestions formulades pels membres titulars del tribunal, aquest atorga la qualificació:

☐ NO APTE ☐ APROVAT ☐ NOTABLE ☐ EXCEL·LENT

(Nom, cognoms i signatura) President/a	(Nom, cognoms i signatura) Secretari/ària

(Nom, cognoms i signatura) Vocal	(Nom, cognoms i signatura) Vocal	(Nom, cognoms i signatura) Vocal

______________________, _______ d'/de __________________ de _______________

El resultat de l'escrutini dels vots emesos pels membres titulars del tribunal, efectuat per l'Escola de Doctorat, a instància de la Comissió de Doctorat de la UPC, atorga la MENCIÓ CUM LAUDE:

☐ SÍ ☐ NO

(Nom, cognoms i signatura) President de la Comissió Permanent de l'Escola de Doctorat	(Nom, cognoms i signatura) Secretari de la Comissió Permanent de l'Escola de Doctorat

Barcelona, _______ d'/de ____________________ de _________

Acknowledgements

I would like to express my gratitude to my advisors, Oscar Esparza and Juanjo Alins, whose expertise, understanding, and patience, added considerably to my graduate experience. I appreciate their vast knowledge and skills in many areas, and their assistance in writing reports. Above all and the most needed, they provided me invaluably constructive criticism and friendly advice during the project work.

Second, and always be grateful to Miquel Soriano for giving me the opportunity to be part of the Information Security Group, and for helping me in the fulfillment of this thesis.

I am also deeply indebted to Jorge Mata and Jose L. Muñoz. Without their guidance, support and good nature, I would never have been able to develop this thesis successfully. I benefited greatly from their ideas and insights. I would also like to thank to Ernesto Exposito and Mathieu Gineste, to have given to me the possibility to work with them at CNRS-Toulouse.

Some debts are hard to put into words. My research colleagues Carlos Gañán, Juan Caubet and Javi Parra, all know why their names are here.

Finally, I would like to dedicate this thesis to Lucinda and my parents. Lucinda, my travel companion, for being by my side through good and bad moments, always ready to listen to me, encourage me and support me. My parents, for everything, but especially for believing in me. Without all of you, none of this would ever have been possible.

I realize that not all people who contributed either directly or indirectly to my study are mentioned in this page. I would also like to thank all of you...

Abstract

Intelligent Transportation Systems (ITS) has impulsed the vehicular communications at the present time. The vehicular communications field is a hot research topic and is attracting a great interest in the automotive industry and telecommunications. There are essentially two main lines of work: (1) communication services related to road safety and traffic information; and (2) information and entertainment services, also named *infotainment* services. These latter services include both transmitting multimedia (voice over IP, streaming, on-line gaming, etc.) and classic data services (e-mail, access to private networks, web browsing, file sharing, etc.). In this thesis we will focus on these infotainment services because further research in this immature research field is necessary and, until nowadays, the main effort of the research community regarding vehicular communication has been focused on road safety and traffic information.

Vehicular nodes need to be reached from the Internet and vice versa to be able to access to infotainment services. While vehicles move along the road infrastructure, they change their wireless point of attachment to the network. During this process, connectivity breaks down until the vehicle is connected again to a new road side unit in its area. This disconnection causes a disruption in the communications. Fast handoffs are a crucial requirement for vehicular networks to avoid long disruption times, since the high speed of vehicular nodes involves suffering a lot of handoffs during an Internet connection.

This thesis is focused on Vehicular-to-Infrastructure (V2I) real-time infotainment services. The main contributions of this thesis are: i) a new testing framework for V2I communications to be able to test infotainment services in an easy way; ii) the analysis of the deployability of infotainment video services in vehicular networks using mobility protocols; and iii) the development of a new Transport Control Protocol (TCP) architecture that will provide a better performance for all TCP-based infotainment services in a vehicular scenario with handoffs.

In this thesis, firstly, we propose a new testing framework for vehicular infotainment applications. This framework is a vehicular emulation platform that allows testing real applications installed on Linux virtual machines. Using emulation, we are able to evaluate the performance of real applications with real-time requirements, so we can test multimedia

applications used to offer infotainment services in vehicular scenarios in a straightforward way.

Secondly, using the testing framework implemented in the first part of the thesis, we have done a performance evaluation of an infotainment service. Among these services, we think that video on demand services on highways will be interesting for users, and generate revenue to network operators. So we evaluated how network-layer handoffs can limit the deployment of a video streaming service. According to the results obtained, driving at high speeds will be an issue for a correct playback of video content, even using fast handoffs techniques.

Finally, we developed a new TCP architecture to enhance performance during handoffs. Most of the *infotainment* services on ITS rely on TCP, one of the core protocols of the Internet Protocol Suite. However there exists several issues related to TCP and mobility that can affect to TCP performance, and these issues are particularly important in vehicular networks due to its high mobility. Using new IEEE 802.21 MIH services, we propose a new TCP architecture that is able to anticipate handoffs, permitting to resume the communication after a handoff, avoiding long delays caused by TCP issues and adapting the TCP parameters to the new characteristics of the network. Using the architecture proposed, the performance of TCP is enhanced, getting a higher overall throughput and avoiding TCP fairness issues between users.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Context

Intelligent Transportation Systems (ITS) have impulsed the vehicular communications at the present time. The vehicular communications field is a hot research topic and is attracting a great interest in the automotive and telecommunications industry. There are essentially two main lines of work: (1) communications services related to road safety and traffic information; and (2) information and entertainment services, also named infotainment services. These latter services include both transmitting multimedia (voice over IP, streaming, on-line gaming, etc.) and classic data services (e-mail, access to private networks, web browsing, file sharing, etc.).

Also, two types of communications are defined:

- V2V (Vehicle-to-Vehicle). This type of communication is between vehicular nodes, through multi-hop or through a single hop, but without the participation of an infrastructure, using *ad hoc* networks.

- V2I (Vehicle-to-Infrastructure) or I2V(Infrastructure-to-Vehicle). This type of communication involves both the vehicular nodes and the infrastructure (e.g. Internet communications).

Vehicular networks have been primarily driven for safety reasons. However, non-safety applications are also important for the successful deployment of them, mainly because infotainment applications will probably be an impulse not only for users, but also for network operators because they will be an interesting business opportunity that will promote the necessary investment in a road-side infrastructure.

Notice that the communication requirements for safety and non-safety applications are very different. For example, safety applications usually disseminate data in geographical areas. This strategy results in unique protocol mechanisms for geographically-based data forwarding, congestion control, and reliable data transfer with strong cross-layer dependencies. Usually, these mechanisms are not part of the TCP/IP protocol stack. On the other hand, non-safety applications establish sessions with other principals, and the data dissemination strategies depend on each particular application.

Despite this, safety and non-safety applications will probably be integrated into a single system. There are several initiatives that try to standardize vehicular communications and integrate both (V2V and V2I) types of communications in a single protocol stack. For example, the Car-to-Car Communication Consortium (C2C-CC), has defined a Car-to-Car Communication (C2C-C) [39] protocol stack that offers specialized functionalities and interfaces to applications. The IEEE is also involved in the standardization of vehicular communications with the standard family IEEE 1609 - Wireless Access in Vehicular Environments (WAVE) [4]. Another relevant project for vehicular communications is leaded by the European Telecommunications Standards Institute (ETSI) and is aimed to develop the CALM concept (Continuous Air-Interface Long and Medium Range) [1]. The goal of CALM is to develop a set of standards to get seamless communications in vehicular networks using different access networks and different technologies.

All these initiatives use WAVE physical (PHY) and medium access control (MAC) layers based on IEEE 802.11p [48] as access technology, which is an amendment to the IEEE 802.11 standard to enhance the wireless access in the vehicular environment. IEEE 802.11p is currently considered the best candidate for basic safety-oriented systems, and it is allocated around 5.9 GHz in a protected frequency band dedicated to road safety. C2C and ETSI also consider that other types of data traffic may rely either on different frequency bands or on alternative wireless technologies. In particular, one or more amendments of the 802.11 standard (i.e., IEEE 802.11a/b/g/n), or 3GPP technologies (i.e., UMTS or LTE), can also be used in vehicular networks with minimum additional complexity.

One of the main problems in vehicular communications is produced by the high mobility of vehicles, which generate network partitions frequently, and this is normally translated to a lack of route availability, causing disruptions and packet loss. These mobility and route availability problems are addressed by ad hoc routing protocols when only V2V communications are used [74].

On its side, in a V2I communications scenario most infotainment services require Internet access, so an Internet gateway is needed. Vehicles cannot be attached to the network using a static point due to their mobile nature. A global addressability and bidirectional Internet connectivity is needed. Mobility management protocols should guarantee global reachability and seamless mobility of nodes in the vehicular network. A network layer mobility solution like Mobile IP [88] can be used to provide access to vehicles to the Internet. Also, Network Mobility (NEMO) [45] can be used when each vehicle is considered as a whole network that moves, permitting that the passengers' devices can be plugged into the car communication equipment. These mobility protocols are basic to provide seamless connectivity to a peer that is moving between different subnets or domains. Fast handoffs are also a crucial requirement for vehicular networks due to the high speed of vehicular nodes. This way, Fast Handoffs for Mobile IP (FMIP)[1] [71] should also be considered.

In this thesis we will focus on infotainment services, because further research in this immature research field is necessary. Until nowadays, the main efforts of the research community regarding vehicular communication have been focused on road safety and traffic information. Also, we will focus our efforts on V2I communications, emphasizing the aspects related to mobility issues and handoff management. In this spirit, it is clear that any optimized mechanism aimed at achieving seamless communications during vehicular trips will enhance users' quality of experience when using Internet based services based on V2I communications.

1.2 Objectives

This thesis aims to mitigate the issues of mobility for V2I communications used for infotainment services over vehicular networks. The work of this thesis follows three lines: (1) the design and implementation of a vehicular simulator based on emulation to provide a testing tool for real-time infotainment services, such as video based services over vehicular networks; (2) a performance evaluation of video services using Mobile IP and FMIP protocols; and (3) a solution to enhance the TCP communications handoff procedures using IEEE 802.21 cross-layer information. Therefore, the objectives of this thesis are as follows:

1. **Analysis of existing vehicular simulation tools:** Researchers and developers need a framework to evaluate protocols and services in this challenging scenario. But preparing and performing tests in real scenarios can be extremely costly and several drawbacks can appear due to the difficulty of managing a fleet of cars. For this reason, software experiments can play an important role to test vehicular scenarios, and in fact most research in vehicular networks relies on simulations. Network simulators combined with traffic models generated by mobility simulators can recreate both the vehicular network and the mobility pattern. In this thesis we analyze existing simulation tools that can be used to test infotainment services in vehicular networks.

[1] FMIP is an enhancement of the Mobile IP protocol to get seamless communications during handoffs

2. **Design and implementation of a vehicular emulator platform:** Most conventional simulators are unable to simulate networks in real-time, so the results obtained during these simulations can vary from the real behavior. Another typical inconvenience is that most existing vehicular simulation platforms are focused on V2V communications for safety applications. This means that they simulate the ad hoc domain pretty well, allowing communications among cars using multi-hop and V2V communications, but they do not consider the infrastructure side, which is essential for most infotainment services based on V2I communications. For this reason, we propose a testing framework for infotainment applications called VESPA (Vehicular EmulationS Platform for real Applications), that consists of a set of software developments and a GUI tool that integrates a network simulator with emulation features (ns-2) [9], a road traffic mobility simulator (SUMO) [36] and UML (User Mode Linux) virtual machines [46].

3. **Evaluate the performance of video based services using V2I communications:** Infotainment services are becoming more and more attractive to users, and more particularly, video streaming applications, which can provide services like video on demand or road-side video advertisement broadcasting. However, video streaming applications under vehicular networks suffer from playback disruptions resulting from handoff blackout periods. Despite the importance of reliable results, nearly all ongoing research activities addressing video streaming over vehicular networks are based on V2V communication simulation studies that neglect the effects of frequent handoffs over real video applications. We use VESPA to study the performance of video infotainment applications with infrastructure participation in vehicular networks. We present a study for the potential deployment of video on demand services in vehicular networks where a Mobile IP solution is used for real-time video using UDP+RTP protocols. In this study we gauged the effects of mobility over the video transmission using Mobile IP and Fast Handovers for Mobile IP (FMIP) protocols. We show that although fast handoffs techniques minimize blackouts using slow speeds, the recurrence of handoffs at high speeds limits the deployment of video streaming services in vehicular networks.

4. **Propose a TCP modification to alleviate the impact of infrastructure handoffs in V2I communications:** There are some issues related to the use of the Transmission Control Protocol (TCP) in vehicular networks. TCP is a protocol designed for the wired network and reacts to packet loss caused by handoffs as a signal of network congestion, dropping its congestion window (cwnd) and reducing the transmission

rate. Old TCP states can cause poor performance after reconnection due to a bad configuration of TCP timers, and because TCP needs some time to learn new parameters. Also, handoffs disruptions can cause unfairness between vehicular nodes going at different speeds. We propose a new architecture, named VSPLIT, for V2I communications to enhance the handoff procedures when using TCP in IEEE 802.11 networks, which is based on the IEEE 802.21 Media Independent Handover (MIH) services. The proposed architecture uses a new version of TCP that we developed. This new version of TCP modifies the standard congestion control, learning the characteristics of the new network after the handoff, and using the cross-layer information provided by the MIH services. Our architecture is a TCP-splitting architecture where the modified TCP protocol is used between a Performance-Enhancing Proxy (PEP) and the vehicular user. The use of PEPs allows Internet hosts to use standard TCP. VSPLIT architecture reduces the handoff disruption time for TCP communications during handoffs, increases the aggregated throughput of all the vehicular users in the network and enhances the fairness between TCP connections in the vehicular network.

1.3 Related Publications

This thesis has been supported partially by the Spanish Research Council with Project TEC2011-26452 (SERVET), by Spanish Ministry of Science and Education with Project CONSOLIDER CSD2007-00004 (ARES), by Generalitat de Catalunya with Grant 2009 SGR-1362 to consolidated research groups and with the support from the SUR (Secretaria d'Universitats i Recerca) of the DEC (Departament d'Economia i Coneixement), and by the European Social Funds. Most of the research results presented in this dissertation have been published in journals and conferences. In this section we provide a list of such publications, together with their complete bibliographic information. Further, we include other complementary articles that are not directly related with the research topic of this thesis, but which are especially significant from the state-of-the-art perspective.

Journal publications:

1. Sergi Reñé, Oscar Esparza, Juanjo Alins, Jorge Mata-Díaz, and Jose L. Muñoz. "VSPLIT: A Cross-Layer Architecture for V2I TCP Services Over 802.11", *Mobile Networks and Applications*, vol. 18, no. 6 (December 2013), pp.831-843.
   DOI=10.1007/s11036-013-0473-8 http://dx.doi.org/10.1007/s11036-013-0473-8
   (Impact Factor: 1.496)

2. Sergi Reñé, Juanjo Alins, Jorge Mata-Díaz, Carlos H. Gañán, and Jose L. Muñoz. "Vespa: Emulating Infotainment Applications in Vehicular Networks", *Pervasive Computing, IEEE*, vol.13, no.3, (July-Sept. 2014), pp.58-66, DOI=10.1109/MPRV.2014.60 http://dx.doi.org/10.1109/MPRV.2014.60 **(Impact Factor: 2.103)**

Conference publications:

3. Sergi Reñé, Carlos H. Gañán, Juan Caubet, Juanjo Alins, Jorge Mata-Díaz, and Jose L. Muñoz. "Analysis of Video Streaming Performance in Vehicular Networks", in Proceedings of the *International Conference on Advanced Communications and Computation (INFOCOMP)*, Barcelona, Spain, October 2011, pp. 92-97. ISBN: 978-1-61208-161-8.

Finally, we list the complementary articles mentioned at the beginning of this section.

4. Sergi Reñé, Ernesto Expósito, Mathieu Gineste, Juanjo Alins and Oscar Esparza. "Multipath TCP Architecture for Infotainment Multimedia Applications in Vehicular Networks", in Proceedings of the *the 81st IEEE 81st Vehicular Technology Conference (VTC)*, Glasgow, Scotland, May 2015 (*to appear*).

5. Carlos H. Gañán, Sergi Reñé, Jose L. Muñoz-Tapia, Oscar Esparza, Jorge Mata-Díaz, and Juanjo Alins. "Secure handoffs for V2I communications in 802.11 networks", in Proceedings of the *the 10th ACM symposium on Performance evaluation of wireless ad hoc, sensor, & ubiquitous networks (PEWASUN)*, Barcelona, Spain, October 2013, pp. 49-56. ISBN: 978-1-4503-2360-4 .

6. Carlos H. Gañán, Johnatan Loo, Arindam Gosh, Oscar Esparza, Sergi Reñé, and Jose L. Muñoz. "Analysis of Inter-RSU Beaconing Interference in VANETs", in Proceedings of the *5th International Workshop on Multiple Access Communications (MACOM)*, Lecture Notes in Computer Science (Springer), vol. 7642, Maynooth, Ireland, November 2012, pp. 49-59 ISBN: 978-3-642-34975-1.

7. Carlos H. Gañán, Juan Caubet, Sergi Reñé, Jorge Mata-Díaz, Juanjo Alins, and Jose L. Muñoz. "NeuroCast: Adaptive Multi-source P2P Video Streaming Application for Wireless Networks", in Proceedings of the *Wired/Wireless Internet Communications (WWIC)*, Lecture Notes in Computer Science (Springer), vol. 6649, Vilanova i la Geltrú, Spain, June 2011, pp. 272-284 ISBN: 978-3-642-21559-9.

1.4 Outline of this Thesis

The structure of this dissertation is in line with the research objectives defined in Section 1.2. Chapter 2 illustrates the background of V2I communications. This chapter presents what a vehicular network is, the different standardization initiatives that exist and the different issues in V2I communications that this thesis will deal with. Chapter 3 presents the analysis of existing vehicular simulation tools and the design and implementation of VESPA, a real-time open-source emulation platform that allows testing real implementations of infotainment applications. Chapter 4 focuses on the analysis of the deployability of real infotainment applications in a vehicular network scenario. We evaluate the effects of handoffs between RSUs caused by vehicles mobility. We simulate network-layer handoffs using Mobile IP and FMIP protocols and we analyze the performance of a video playback in a highway scenario with both protocols. Chapter 5 details the new TCP-splitting architecture for vehicular environments to enhance the handoff procedures using a modified TCP protocol is used between a Performance-Enhancing Proxy (PEP) and the vehicular users. Finally, Chapter 6 concludes this thesis summarizing the main findings of the presented work and making suggestions for the future research. In Appendix A we detail the Mobile IP, and the FMIP protocols. In Appendix B we include some guidelines about how to use VESPA.

Chapter 2

Background

2.1 Vehicular Communications and Services

Vehicular networks based on Dedicated Short-Range Communications (DSRC) [65] provide communications among vehicles and the roadside infrastructure. DRSC is a kind of communications, operating in the 5.9 GHz licensed spectrum band, which physical (PHY) and medium access control (MAC) layers are defined in the IEEE 802.11p standard. DRSC support the requirements of vehicular communications, such as achieving high and reliable performance in highly mobile, often densely populated, and frequently non-line-of-sight environments. DSRC involve several entities and different network domains. DSRC entities are depicted in Figure 2.1. Vehicular Nodes (VN) are equipped with devices termed *On-Board Units* (OBU), which implement the communication protocols and algorithms. OBUs can communicate among them, or with fixed stations installed along roads termed *Road Side Units* (RSU). OBUs and RSUs implement the same protocol functionalities and form a self-organizing network, also called as the *Ad-hoc Domain*. OBUs offer an interface to the devices present in the car, which are called *Application Units* (AUs). These AUs and OBU form another mobile domain, which is usually termed *In-Vehicle Domain*. RSUs can either be isolated or attached to a larger structured network. If RSUs are isolated, their function is usually to distribute static information (e.g. dangerous curve, construction site ahead) or simply to extend the OBUs communication range by acting as forwarding entities. If RSUs are part of a large infrastructure deployed along the road, they are usually responsible for assuring connectivity to vehicles. This infrastructure network is generally called the *Infrastructure Domain*.

Applications for vehicular networks are grouped into safety (e.g. hazard warning, work-zone warning) and non-safety applications (e.g. point-of-interest notification, Internet access). These application types put different and partially conflicting requirements on the

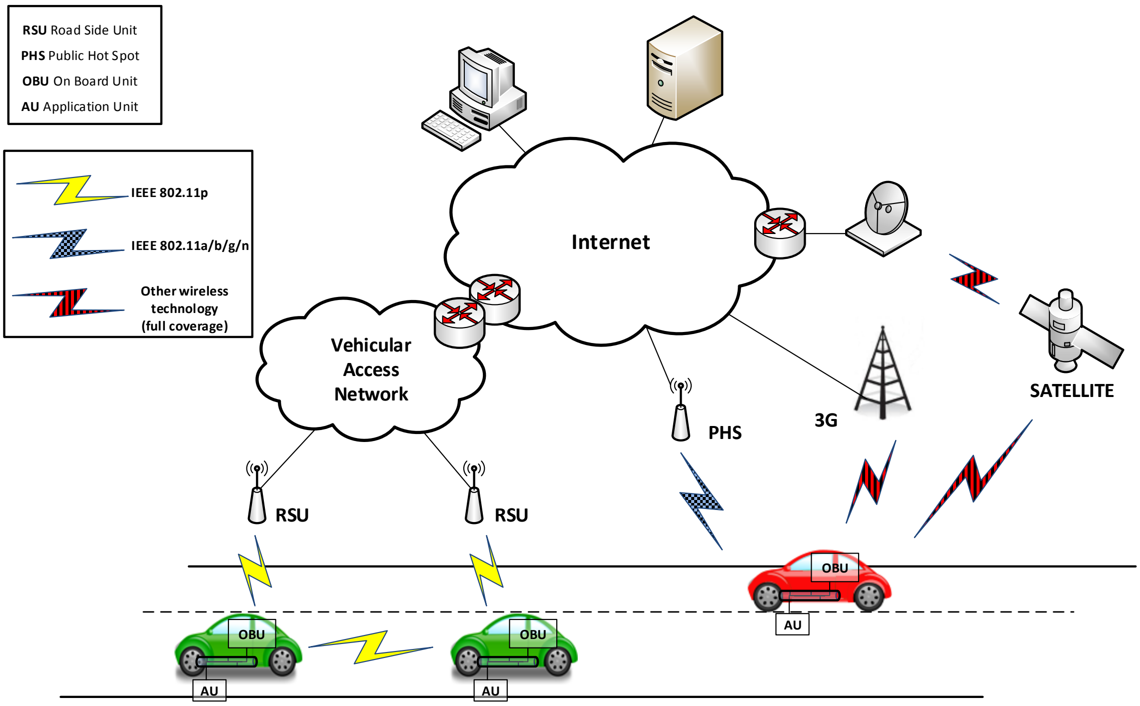

Fig. 2.1 Vehicular Network

system design. The communication requirements for safety and non-safety applications are different. On one hand, non-safety applications typically establish sessions with other peers using the Internet protocols. Data are transmitted as packets from source to destination, using unicast or multicast. On the other hand, safety applications usually disseminate data in geographical areas. This implies in-network processing that allows aggregating, modifying, and invalidating the information to be forwarded. The fundamentally different information dissemination strategy of safety applications results in unique protocol mechanisms for geographically-based data forwarding, congestion control, and reliable data transfer with strong cross-layer dependencies [77]. Usually, these mechanisms are not part of the TCP/IP protocol stack.

In order to reach a considerable number of equipped vehicles after market introduction, safety and non-safety applications must be integrated into a single system. In particular, a number of safety applications need a minimum share of equipped vehicles for vehicle-to-vehicle communication. The support for non-safety applications is also important for successful market introduction of a safety communication system and the successful deployment of a vehicular network infrastructure. DSRC serve as the basis for connected vehicle safety and infotainment applications integration. Infotainment jointly with traffic efficiency applications can improve drivers' experience, making vehicular communications systems more attractive to end-users. Internet access, multiplayer games, multimedia applications, chat and videoconference are examples of infotainment services. Infotainment applications will be an impulse not only for users, but also for network operators because they will be an interesting business opportunity that will promote the necessary investment for infrastructure deployment.

Infotainment services are mainly related to the provision of classic IP applications, using common Internet protocols over IPv6. Connections to the Internet can be established by using V2I communication, allowing typical communication services like web browsing, mail or chat. Infotainment services also can be used by the passengers of a vehicle to be informed of nearby services, restaurants, companies or touristic sights. Some examples of infotainment services are listed in Figure 2.2):

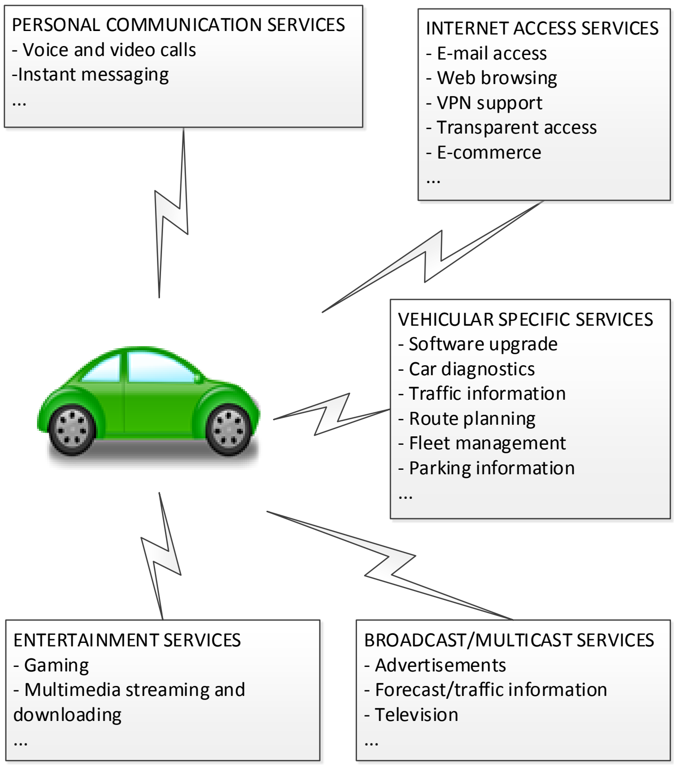

Fig. 2.2 Infotainment services examples

Among all the open issues that are present in the vehicular networks environment, there are three main functionalities that must be provided to V2I communications in order to provide the availability of IP communications to vehicular users: address autoconfiguration, efficient routing and mobility management. These three main aspects that must be addressed in V2I communications were firstly introduced in [26]. Address autoconfiguration [25, 32] and efficient routing [30] are out-of-the-scope of this thesis. This thesis is focused on mobility issues.

Bringing infotainment services to the vehicular environment requires complying with standard protocols and mechanisms that allow heterogeneous networks to be interconnected in the Internet. In the next section we present the most important standard initiatives that propose a protocol stack for vehicular networks.

2.2 ITS Standardization initiatives

It is important the development of an adequate standard compatible either with V2V and V2I communications, supporting both safety and infotainment applications. The standardization of vehicular network protocols concerns to different international organizations. A lot of research projects has led to standardization initiatives from different parts of the world. For example, American research projects, such as Cooperative Intersection Collision Avoidance Systems (CICAS) [43], SafeTrip21 [98] or California Partners for Advanced Transportation Technology (PATH) [83]; Japanese projects, such as Smartway [92] or ITS-Safety 2010 [27]; and European projects, such as CVIS Cooperative Vehicule-Intrastructure System [44], NOW Network-on-Wheels [81] or SEcure VEhicular COMmunication [100]. All these, among others, have served as a basis to develop vehicular standards for the different standardization organizations. The IEEE has developed the protocol stack WAVE, including an extension of the 802.11 family protocols for the low layers, as well as an alternative to IP in higher layers. The Car-to-Car Communications Consortium (C2C-CC) has developed and experimented specific protocols for vehicular networks. The ETSI Technical Committee ITS is involved in the harmonization of ISO, IETF, IEEE and C2C standards.

Next we summarize the main proposed standards to develop a protocol stack for vehicular networks, both from public and private organisms.

2.2.1 IEEE 1609

IEEE has defined WAVE (Wireless Access in Vehicular Environment) or the 1609 protocols family. WAVE specifies a complete protocol stack (1609.0 to 1609.4), relying on 802.11p for the low layers. The DSRC radio technology 802.11p is essentially IEEE 802.11a adjusted for low overhead operations in the DSRC spectrum. The overall DSRC communication stack between the link layer and applications has been standardized by the IEEE 1609 working group. Hence, IEEE 1609 is a higher-layer standard on which IEEE 802.11p is based. Indeed, the IEEE 1609 family of standards for wireless access in vehicular environments consists of four standards:

- IEEE 1609.1 - Resource Manager: It defines the basic application platform and includes application data read/write protocol between RSU and OBU.
- IEEE 1609.2 - Security Services: It defines the 5.9-GHz DSRC security, anonymity, authenticity, and confidentiality services.
- IEEE 1609.3 - Networking Services: It defines network and transport layer services, including addressing and routing, in support of secure WAVE data exchange.
- IEEE 1609.4 - Multichannel Operations: It provides DSRC frequency band coordination and management, where it manages lower-layer usage of the seven DSRC channels, and integrates tightly with IEEE 802.11p.

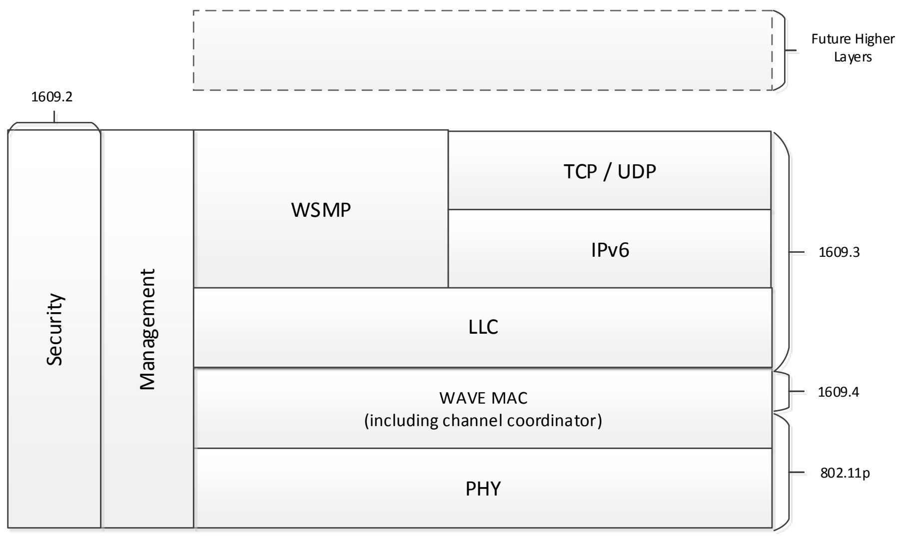

Fig. 2.3 WAVE protocol stack

The WAVE protocol stack is depicted in Figure 2.3. Non-safety applications can use the traditional Internet protocol stack containing IPv6, and the transport layer User Datagram Protocol (UDP) for connectionless services, as well as Transmission Control Protocol (TCP) for connection-oriented services. Both parts of the protocol stack share the same data link layer and physical layer for transmission. The 1609.3 standard includes the WSMP protocol (WAVE short Messages Protocol) for V2V communication, presented as an alternative to IPv6. In this protocol, messages are routed with an Application Class Identifier

(ACID) and an Application Context Mark (ACM) to replace the IP address and the port number. This would ease the communications in dynamic environments. IEEE 1609.2 adds a transversal layer responsible of the security, anonymity, authenticity and confidentiality of security communications. IEEE 1609.4 defines optional multichannel operations to manage the usage of the seven licensed DSRC channels for single radio devices.

2.2.2 C2C

The CAR 2 CAR Communication Consortium (C2C-CC) [39] is a non-profit industrial driven organization initiated by European vehicle manufacturers (Audi, BMW, Daimler-Chrysler, Fiat, Renault, and Volkswagen) supported by equipment suppliers, research organizations and other partners. The C2C-CC is dedicated to the objective of further increasing road traffic safety and efficiency by means of cooperative ITS (V2V) with inter-vehicle communications supported by V2I communications. The C2C-CC supports the creation of a European standard for future communicating vehicles spanning all brands. The C2C-CC also works in close cooperation with European and international standardization organizations, in particular the ETSI TC ITS. The C2C-CC follows the realistic deployment strategy and business model in order to speed-up the market penetration, and it is a roadmap for the deployment of V2V and V2I services

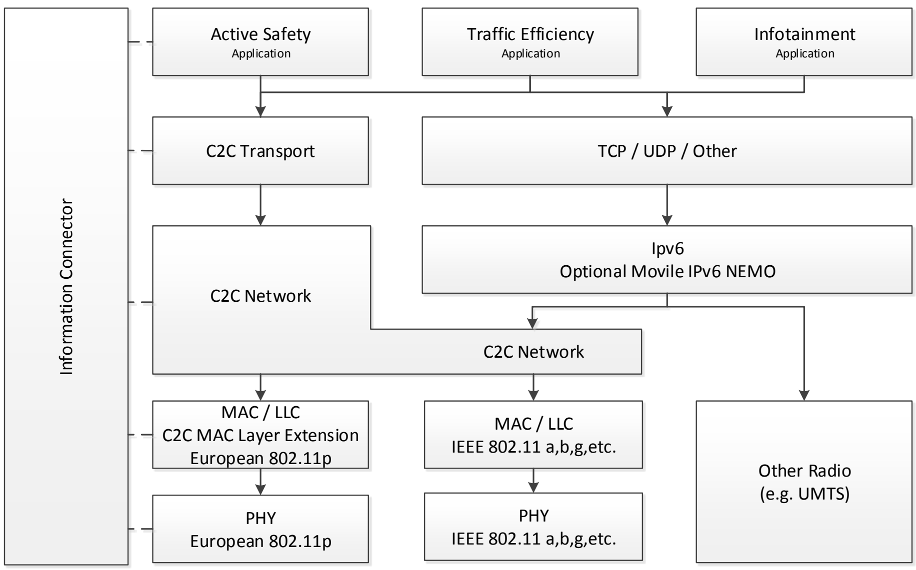

Fig. 2.4 C2C-CC protocol stack

C2C-CC defines a protocol stack (Figure 2.4) based on Geocast [30] ad hoc protocol for V2V communications and Mobile IP/NEMO for V2I communications. Geocast routing protocols rely on GPS positions to route messages from vehicle to vehicle. Vehicular communications in C2C-CC are also based on DSRC, which rely on the IEEE 802.11p protocol. However, in contrast with WAVE standard, also takes into account optionally other access technologies, such as WLAN 802.11 protocols (802.11a/b/g/n) or 3GPP technologies.

The consortium is looking forward to allowing interoperability among cars from different car manufacturers and suppliers of on-board and roadside units. In this context, the C2C-CC is concerned with real-life demonstrations of safety applications for tangible ad hoc networks, providing a framework for system prototyping. C2C-CC demonstrates the C2C-System as proof of technical and commercial feasibility.

The C2C-CC is well connected to other organizations. Various European R&D projects contributed to specifications of C2C. There is a close cooperation between C2C-CC and ERTICO [91] (an European public-private partnership for ITS) that ensures the deployment of the developed standards, and interaction with other standardization development organizations (CEN, CENELEC, IEEE, ISO, ITU).

2.2.3 ETSI TC ITS

The European Telecommunications Standards Institute (ETSI) is developing a unified ITS architecture for Europe upon the CALM (Continuous Air-interface Long and Medium) concept and the C2C architecture, detailed in the previous section. The ETSI TC ITS station reference architecture [24] is depicted in Figure 2.5. This architecture was firstly described in the COMeSafety Project [90] in 2008. The International Organization for Standardization (ISO) [60] also fulfills this CALM architecture. The concept of CALM is based on heterogeneous cooperative communication framework to provide continuous communication to vehicular nodes.

Several communication layers are defined in the architecture: applications layer on top, followed by a facilities layer and the networking and transport layer. Below, the access technologies layer is placed, where again multiple communication technologies may be used. Apart from the transversal management plane, the architecture also defines a layer-independent security plane. The facilities layer is capable to provide the basic services that are common for all applications and it bundles information that different applications want to transmit. As an example, positioning information is only contained once in the transmitted messages, but may be used by several applications. The ETSI ITS-G5 protocol (ETSI ES 202 663), an adaption of IEEE 802.11p, has been defined to be used in direct communication between vehicles in Europe. The access layer combines the data link layer and the physical layer and it is perceived as a single entity. The security plane can be viewed

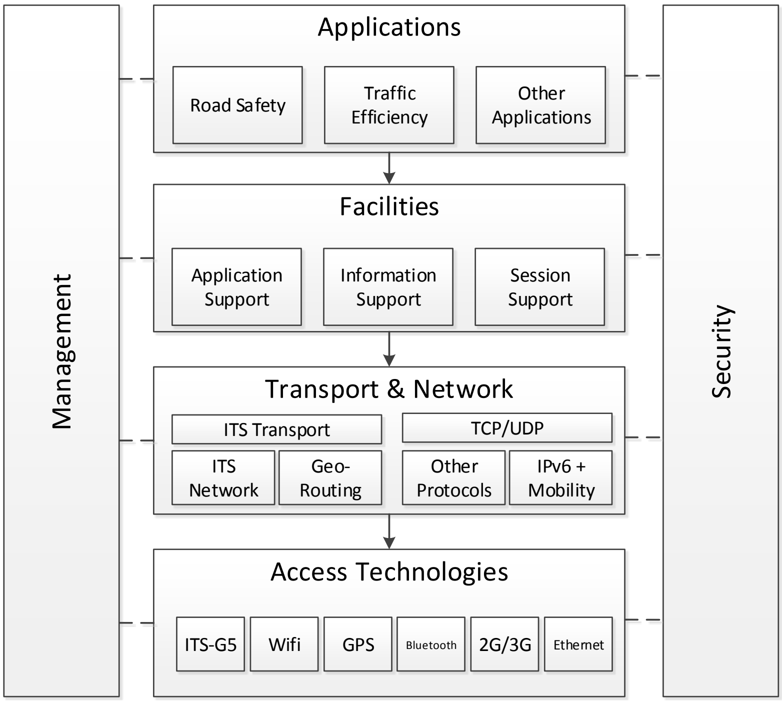

Fig. 2.5 ETSI protocol stack

as a specific part of the management plane. The network and transport layers are grouped together in a similar way as in the WAVE approach in Fig. 2.3.

2.3 IEEE 802.11p

All the previous standardization initiatives detailed in Section 2.2 use PHY and MAC layers based on IEEE 802.11p [22] as one of the main candidate access technologies. The IEEE 802.11p standard is an amendment to the IEEE 802.11 standard to enhance the wireless access in the vehicular environment, and it is currently considered the best candidate for DSRC communications.

The purpose of 802.11p is to provide the minimum set of specifications required to ensure interoperability between vehicular wireless devices. This is due to IEEE 802.11p devices may be used in environments where the physical layer properties are rapidly changing and where very short-duration communications exchanges are required, for instance in situations where transactions must be completed in less time frames than the minimum possible in IEEE 802.11, either using infrastructure or ad hoc mode. 802.11p is based on extensive testing and analyses of wireless communications in a mobile environment. This previous work that defines a MAC and PHY for DSRC, based on 802.11 and 802.11a technologies, is documented in the ASTM E2213-03 standard [23] and it is technically compatible with the 802.11p amendment.

IEEE 802.11p allows a Vehicular Node (VN) that is not a member of a Basic Service Set (BSS) to transmit data frames. The BSS provides the basic building-block of an 802.11 wireless LAN, and consists of an access point and its associated stations. In 802.11, a node needs to be part of a BSS to be able to transmit data frames. IEEE 802.11p defines a new parameter *dot11OCBEnabled* that allows immediate communication. A VN is able to transmit a data frame outside the context of a BSS only if the defined *dot11OCBEnabled* parameter is true. This avoids the latency associated with the IEEE 802.11 authentication, association, or data confidentiality services required to establish a BSS. When *dot11OCBEnabled* is true, a data frame can be sent to either an individual or a group destination MAC address. Since the IEEE 802.11 MAC sublayer authentication services are not used when *dot11OCBEnabled* is true, any required authentication services would be provided by the Station Management Entity (SME) or by applications outside of the MAC sublayer. This is why all the vehicular standardization initiatives include a security data plane into the protocol stack.

IEEE 802.11 protocol uses the licensed 5.9 GHz frequency band. The PHY layer used in 802.11p is the OFDM physical layer detailed in the IEEE 802.11 standard [19]. The basic idea is to divide the available frequency spectrum into narrower subchannels (subcarriers). The high-rate data stream is split into a number of lower-rate data streams transmitted simultaneously over a number of subcarriers, where each subcarrier is narrow banded. There are 52 subcarriers, where 48 are used for data and 4 are pilot carriers. The OFDM PHY layer supports three different frequency channel widths; 5 MHz, 10 MHz, and 20 MHz. 802.11p is using 10 MHz channels whereas 802.11 access points usually use 20 MHz channels. The OFDM symbol duration and subcarrier frequency spacing are depending on channel widths, i.e., the number of subcarriers is fixed. The duration of one OFDM symbol in 802.11p is 8 μs including guard interval. OFDM has support for eight different transfer rates, which are achieved by using different modulation schemes and coding rates. In Table 2.1 the different transfer rates together with the coding schemes used in 802.11p are shown for 10 MHz frequency channels. Support of 3, 6, and 12 Mbit/s transfer rates is mandatory.

Vehicular denseness will vary from very dense urban areas to sparse highways. Therefore, the MAC layer of a vehicular network must be scalable. The MAC algorithm deployed by 802.11p is found in the 802.11 standard [19] and it is called Enhanced Distributed Coordination Function (EDCA). It is based on the basic Distributed Coordination Function (DCF) of the 802.11 standard, but adds QoS attributes.

DCF employs a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)

Table 2.1 Transfer rates, modulation schemes and coding rates found in OFDM when using 10 MHz channels

Transfer rate (Mbit/s)	**Modulation scheme**	**Coding Rate**	**Data bits per OFDM symbol**	**Coded bits per OFDM symbol**
3	BPSK	1/2	24	48
4.5	BPSK	3/4	36	48
6	QPSK	1/2	48	96
9	QPSK	3/4	72	96
12	16-QAM	1/2	96	192
18	16-QAM	3/4	144	192
24	16-QAM	2/3	192	288
27	16-QAM	3/4	216	288

with binary exponential backoff algorithm. In CSMA/CA a station starts by listening the channel before a transmission and, if the channel is perceived as free for a predetermined listening period, the station can start to transmit directly. If the channel is or becomes occupied during the listening period, the station must perform a backoff procedure, i.e., the station has to defer its access a randomized time period. The predetermined listening period is called Distributed Interframe Space (DIFS) and the selected value for the randomized backoff depends on a contention window. PIFS and SIFS are interframe space values used to give priorities to access points and acknowledgments packets, respectively.

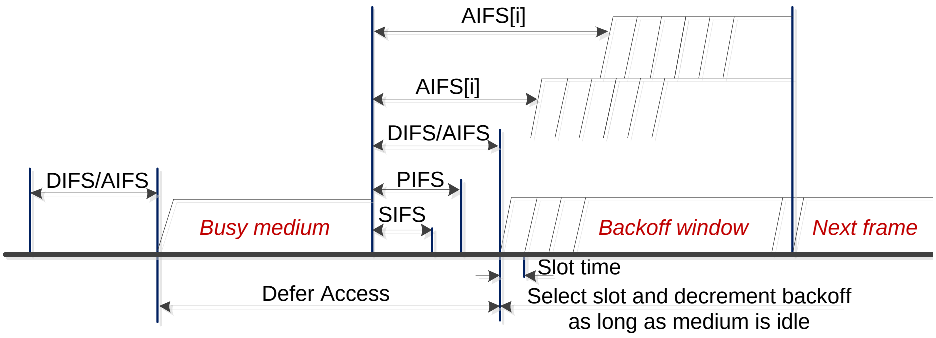

Fig. 2.6 EDCA channel access prioritization, as specified in [20]

EDCA mechanism enhances DCF by using the same prioritization techniques than IEEE 802.11e [20], namely the Hybrid Coordination Function (HCF). In EDCA, service differentiation is provided by assigning different contention parameters to different access categories. EDCA changes DIFS inteframe spaces and a unique contention window for Arbitration Inter-frame Space (AIFS[i]) and different contention windows for different access categories (priority (i)) (see Fig. 2.6). EDCA defines 4 access categories (audio, video, best effort and background).

In unicast mode (one-to-one transmissions), IEEE 802.11 acts as a stop-and-wait protocol; it will await an ACK in return if the message was received successfully. If the ACK is lacking, the transmitter must perform a backoff procedure and later try to retransmit the same message until an ACK is received or the retry counter for this particular message has reached its maximum.

2.4 V2I Mobility Management in Vehicular Networks

One of the main problems in vehicular communications is that the high mobility of vehicles produces network partitions frequently, and this translates into a lack of route availability, causing disruption and packet loss. These route availability problems are addressed in ad hoc routing protocols when V2V communications are used. There exist different proposals of ad hoc routing protocols focused on vehicular networks that try to minimize these ad hoc mobility problems [74]. However, when using V2I communications there are additional problems, which is the case of most infotainment services. The Internet must be reachable to access these services, but cars cannot be attached to the network using a static point. In this sense, an Internet gateway is needed to provide global addressability and bidirectional Internet connectivity to VNs. This Internet gateway can be placed in the RSUs.

Mobility management should guarantee reachability between Correspondent Nodes (CN) (e.g. servers) in the Internet and mobile nodes (vehicular nodes) in the vehicular network. Mobility management protocols, such as Mobile IPv6 (MIPv6) and the NEtwork MObility (NEMO) protocols, are envisioned to implement seamless communications in vehicular networks, and coherent with the standardization initiatives explained previously. Mobile IP is responsible for routing IP datagrams on the Internet, independently of the location of the mobile node. Each mobile node is identified with a "home address", which does not depend on the current location of the mobile node. While away from its "home" network a mobile node has two addresses associated: a "care-of address", which identifies its current location; and a "home address", which is provided by its "home network" and it is used as a global identifier. Mobile IP specifies how a mobile node registers with its home agent and how the home agent routes datagrams to the mobile node through a tunnel. NEMO is an extension of Mobile IP, and it allows session continuity for a whole network that is moving. This permits to have a private network inside the vehicle, where the passengers' devices can be plugged into the car communication equipment, which will manage the mobility as a whole network transparently to the passengers' devices.

There also exist protocols that provide faster handoffs than Mobile IP or NEMO, named micromobility protocol. Micromobility protocols minimize network disruptions during handoffs using link-layer information and they assure a minimum QoS for delay-sensitive infotainment applications. Fast handoffs are a crucial requirement for wireless networks with small coverage area, since the vehicle spends only short periods of time at each point of attachment. One of these micromobility protocols is Fast Handoff for Mobile IP (FMIP) protocol [71, 72]. Mobile IP and FMIP protocols are thoroughly explained and detailed in Appendix A.

Mobility protocols were designed to provide continuous connectivity to a peer that is moving between different subnets or domains. However, the use of these mobility protocols can adversely affect upper layers. For instance, Mobile Internet was not designed with video requirements in mind and therefore video infotainment applications may be handled inefficiently in vehicular networks. Video applications and video content are expected to be a growing infotainment service, such as videoconference, real-time traffic information broadcasting or various on-road video entertainments (live sports, news, etc.). The popularity of video streaming has been considerably increased in the last decade and recent studies have shown that video streaming is responsible for 25-40% of all Internet traffic. Usually, the transport protocol for real-time traffic is UDP, mainly due to TCP is well known to be unsuitable for delivering such type of traffic. However, most popular videoconference or video streaming services (Skype, YouTube, Netflix, etc) [35, 94] use TCP as transport protocol due to some known problems of UDP (saturation, firewalls, etc.). On its side, TCP was mainly designed for wired communications, and therefore suffers from a bad performance when there is intermittent connectivity, for instance caused by handoffs.

For these (and many other) reasons the vehicular scenario is so challenging. The high mobility of vehicles creates frequent handoffs, which may result in significant blackouts in the communications and packet losses. To ensure continuous seamless services to TCP-based applications, we need to provide session continuity when changing the access network. To clarify this, in the rest of this section we explain these TCP issues caused by mobility in V2I communications. We also summarize the IEEE 802.21 protocol, which facilitates sharing information between independent network layers or different network entities. In this sense, IEEE 802.21 can be used to support algorithms that enable seamless handoffs between networks of the same type, as well as between different network types.

2.4.1 TCP issues in V2I Communications

TCP is particularly affected by the occurrence of handoffs. Here we enumerate some of the most important issues that TCP suffers in V2I communications:

1. TCP misinterprets any loss during a handoff in V2I communications as a congestion signal. Therefore, TCP will drop its congestion window and, as a consequence, the transmission rate.

2. TCP does not detect when the wireless link is available again after a handoff, and it must wait until a Retransmission TimeOut (RTO) event to restart sending packets.

3. TCP computes the RTO on the basis of measured round-trip time (RTT) values. A sudden significant variation of the RTT after a handoff can lead to two undesired dynamics:

   - If the new link has a larger RTT, the calculated RTO can expire hastily, and therefore TCP drops its congestion window to a minimum value, when not necessary.

   - If the new link has a shorter RTT, the calculated RTO can produce an elapse time (equal to the old RTO) before the TCP sender can recover from a packet loss [104].

4. When the new link is established after the handoff procedure is completed, TCP state variables, which regulate the transmission rate, are either still tailored for the old link or dropped after loss detection, and a long time may be required before retrieving the optimum settings.

5. TCP handoffs performed by vehicular users going at different speeds can produce unfairness behaviors. Those nodes which stay more time connected to the same RSU (slow vehicles) get more throughput, since they suffer less handoffs. Moreover vehicles at high speed may have not enough time to get the congestion window at the correct working point.

6. TCP congestion window permits sending more packets than desired because an increase of the MAC service time (the time needed to send a packet in the MAC layer) implies an increase of RTT. This behavior can overload the network because this RTT increase does not imply an increase in the capacity of the network.

TCP issues are caused not only due to the handoff latency time, but also due to variations in RTTs and in Bandwidth-Delay Products (BDPs) [104], as we can see in the list above. BDP is a well-known concept in measuring the capacity of a "network pipe". The BDP is generally defined as the minimum number of packets (or bytes) in flight of a TCP connection to fully exploit the available link resources, that is:

$$BDP(bytes) = BW(bytes/s) \times RTT(s) \tag{2.1}$$

where BW is the available bandwidth, i.e. the TCP flow's share of bandwidth at the bottleneck of the network, and RTT is the round trip time. TCP congestion window (*cwnd*) indicates the maximum amount of data that can be sent out on a connection without being acknowledged and determines the number of bytes that can be outstanding at any time. To maintain a link fully exploited, the *cwnd* of TCP senders must be set to the BDP. When there is no competing traffic, the TCP flow should be able to obtain all the bandwidth at the bottleneck link.

In a vehicular scenario, we assume that the bottleneck is the wireless vehicular access network. We can assume this because, on one hand, the wireless domain tends to offer inferior performance than wired domain due to bandwidth restrictions and medium contention issues. On the other hand, we assume end-to-end vehicular services are provided by, either local or via Internet, centralized servers with enough resources. In the IEEE 802.11 MAC layer protocol, the sender has to contend to send only one data packet (and get an acknowledgment back) before contending for the channel again. This is clearly very different from the behavior of wired networks, where multiple packets can be sent without waiting for them to reach the other end of the link and being acknowledged. Therefore, the real or effective BDP of a TCP connection in an 802.11 network is smaller than in wired networks due to contention waiting-times. However, when the MAC service time increases due to contention waiting-time, the calculated RTT by the TCP sender also increases, and therefore the *cwnd* permits sending more packets filling the BDP calculated using Equation 2.1. This behavior can overload the network because this RTT increase does not imply an increase in the capacity of the network. In Chapter 5 we will see how we can calculate an effective BDP that will tighten better to the real capacity of an 802.11 network.

There can be problems in the performance of TCP-based applications (related with changing RTTs) when a handoff occurs between links due to unexpected changes in the BDP. This not only happens in vertical handoffs (e.g., handoffs between 802.11 links and satellite links), but also in horizontal handoffs between 802.11 links. This can be caused by differences in the MAC service time (time to gain access to the share) and in the bandwidth available per user, maybe produced by changes in the number of users contending in a collision domain after a handoff.

The characterization of the MAC service time in saturated IEEE 802.11 DCF networks depends on the number of users contending in the same collision domain [105]. To show the

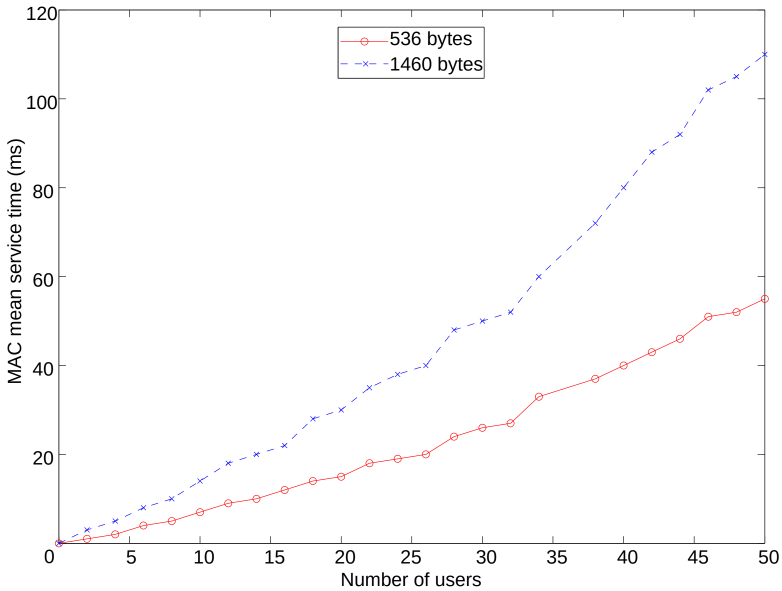

Fig. 2.7 IEEE 802.11 DCF MAC service time

influence of the number of users in the MAC service time we performed some simulations in a simple scenario. Figure 2.7 shows the average MAC service time of an IEEE 802.11 DCF network as a function of number of nodes for packet payloads of 536 and 1460 bytes. We obtained these results by simulating an access point with contending users using ns-3 simulator [10]. The parameters used to setup this simulation are showed in table 2.2. This set of parameters has been chosen because they represent the standard behavior defined in the IEEE 802.11p amendment [22]. Note that RTS/CTS (Request to Send / Clear to Send), an optional mechanism used by the 802.11 protocol to reduce frame collisions introduced by the hidden node problem [64], is enabled.

According to the Figure 2.7, we can observe that the MAC service time increases with the number of users colliding in the same 802.11 channel. Also, the MAC service time rises up more quickly when packet payload is higher. This behavior is caused, as we commented before, because the contention waiting-times are higher when the number of users in a collision domain increases.

2.4.2 IEEE 802.21 Media Independent Handoff (MIH)

Handoff procedures are essential for a good performance of V2I communications in vehicular networks. Handoff procedures are an issue by itself (disruptions during connection migrations) but also generates different issues to TCP protocol. Using multiple network knowledge and cross-layer information we can minimize handoff disruptions avoiding scanning

Parameter Name	Value
PHY	16-QAM 3/4 Code Rate 27 Bytes/Symbol
Slot time	13 μs
SIFS	32 μs
DIFS	58 μs
RTS/CTS	Enabled
Rx Threshold	-82 dBm
CS Threshold	-86 dBm
Tx Power Level	35 dBm
Data rate	27 Mbps
Basic rate	3 Mbps

Table 2.2 MAC and PHY 802.11 parameters

times and discovering available networks in advance [69]. We can also use this information to avoid TCP issues. In Chapter 5 we use the IEEE 802.21 Media Independent Handoff (MIH) standard [21] to provide cross-layer information to the proposed TCP-splitting architecture that enhances TCP performance during handoffs.

The IEEE 802.21 standard [21] defines a media independent entity that provides a generic interface between the different link layer technologies and the upper layers. The main goal of the IEEE 802.21 standard is to facilitate handoffs. This includes handoffs between IEEE 802 and other networks, whether or not they are of different media types (including both wired and wireless), even where handoff is not otherwise defined. IEEE 802.21 also can help mobile devices to perform seamless handoffs where the network environment supports it. These mechanisms are also usable for handoffs between IEEE 802 networks and non IEEE 802 networks. Figure 2.8 represents the MIH framework. IEEE 802.21 defines a logically shim layer, named MIH Function (MIHF), between the link-layer and the network-layer in the protocol stack (see Figure 2.8). IEEE 802.21 allows higher layers to interact with lower layers using the MIHF through a unified interface. These upper layers act as MIHF users and are provided by the services exposed by the MIHF. These MIHF services may be either local or remote, i.e. local operation occurring within a protocol stack and remote operation occurring between two distant MIHF entities.

The MIHF defines three main services:

- Media Independent Event Service (MIES): The MIES provides link layer events to the MIHF treated as discrete events. Event notifications are generated asynchronously.

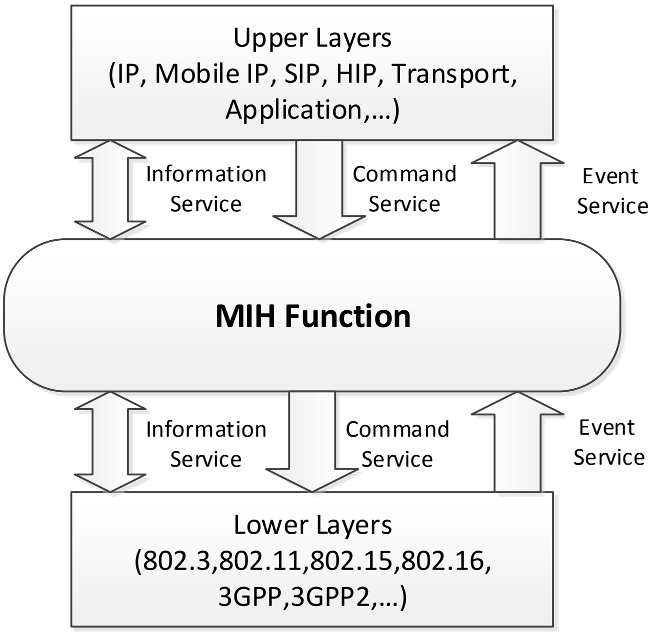

Fig. 2.8 IEEE 802.21 MIH framework

Thus, all MIH users and MIHFs that want to receive event notifications need to subscribe to particular events.

- Media Independent Command Service (MICS): The MICS refers to the commands sent from the higher layers (MIHF users) to the lower layers to determine the status of links or control and configure the MN to gain optimal performance or facilitate optimal handoff policies. Commands are classified into MIH Commands and Link Commands depending on whether these commands are sent by the higher layers to the MIHF or by the MIHF to the link layer.

- Media Independent Information Service (MIIS): The MIIS provides a framework to acquire network information within a geographical area to facilitate handoffs. The MIIS is provided by the Information Server (IS). The IS contains information that is used for network intelligence purposes. The main goal behind the MICS is to allow MN and network entities to discover information that influences the selection of appropriate networks during handoffs. This Information Service provides mostly static information, such as network configuration parameters.

Communications between the MIHF and other functional entities such as the MIHF users and lower layers are based on a number of defined service primitives that are grouped in Service Access Points (SAPs): (1) MIH_SAP allows communication between the MIHF layer and higher-layer MIHF users, (2) MIH_LINK_SAP is the interface between the MIHF layer and the lower layers of the protocol stack and (3) MIH_NET_SAP supports the exchange of information between remote MIHF entities.

Chapter 3

Vehicular Emulations Platform for Real Applications

In the context of the challenging vehicular scenario, researchers and developers need a framework to evaluate their protocols and services. Obviously, the most reliable framework would be to perform an outdoor experiment to evaluate how applications behave under real conditions. However, such framework is extremely costly and several drawbacks can rise due to the difficulty of managing a fleet of cars. For this purpose, software platforms can play a vital role to test real world scenarios and most research in vehicular networks relies on simulations.

Network simulators combined with traffic models generated by mobility simulators can recreate both the vehicular network and the mobility pattern. The problem is that it is not easy, in general, to integrate real implementations of applications within these simulators. Also, existing simulation platforms are mainly focused on providing a testing framework for safety infrastructureless applications, so it is difficult to assess V2I infotainment applications. Most conventional simulators are unable to emulate networks in real time, because simulators use discrete events. In this way, a simulator can efficiently execute network events in batch. On the contrary, emulators (or simulators working on emulation mode) use a scheduler that ties event execution with real time. This makes emulators less scalable compared to simulators, but it permits to inject real traffic from a real application in the modeled network using real time. This permits testing real implementations of infotainment applications, specially applications with real time requirements like multimedia. For this reason, we propose VESPA (Vehicular EmulationS Platform for real Applications), an emulation framework for infotainment vehicular applications with infrastructure participation. VESPA and a set of video testing tools developed for the performance evaluation test-bed used in Chapter 4 can be freely downloaded from *http://sourceforge.net/projects/vespa*.

Here we present the design and implementation of VESPA. In contrast with the existing vehicular simulators, VESPA is able to test real applications installed in virtual machines, recreating a vehicular network. VESPA is able to:

- Emulate different vehicular entities using virtual machines to test real applications in real time.
- Control the experimental conditions and configurations for reproducible evaluation across in a wide range of vehicular scenarios.
- Support large-scale evaluations in terms of network size and node mobility, facilitated by a vehicular traffic simulator.

Using VESPA, researchers and developers can just test their infotainment applications in several vehicular scenarios without worrying about things like how to install a network simulator, or how to generate mobility traces. VESPA allows multiple configurations to support both simplistic and also complex scenarios. VESPA can be used, for example, to compare the performance of various codification techniques (or video players) in a controlled vehicular scenario, or the comparison between different video codification techniques. With VESPA, it is possible to test applications using the same software developed for desktop computers without lasting time in modeling these applications for network simulators, avoiding the limitations caused by simplified application behaviors. Moreover developers can test their software in a complex mobile scenario in a straightforward and fast manner. VESPA provides all the benefits of an emulation tool and, at the same time, it allows using realistic mobility models. VESPA is able to test applications installed in virtual machines using common operating systems (Linux systems). Vehicular entities can be represented by using User Mode Linux (UML) [46] virtual machines. The emulated network in VESPA is obtained using the emulation features of the widely known ns-2 [9] network simulator. The traffic mobility of vehicular nodes is modeled using the Simulation of Urban MObility (SUMO) [36] tool.

VESPA platform is also useful to compare the behavior and the performance of different applications using the same vehicular scenario. For example, we can compare two different video players and test their performance in a highly mobile scenario with lots of handoffs to assess robustness in front of buffer starvation. The use of the emulation features of ns-2 allows us to introduce these vehicular nodes in a live network. Emulation is an essential feature for testing real applications, specially applications with real time requirements (multimedia applications).

Finally, the use of SUMO as vehicular traffic mobility simulator allows us to feed the ns-2 emulator with realistic information about node's mobility. Also, unlike existing proposals to simulate vehicular networks, VESPA offers the possibility to simulate RSUs and server nodes in the infrastructure domain, which are essential for the proper working of most infotainment services. As a consequence, VESPA considers all the layers of the TCP/IP and IEEE 802.11p MAC/PHY protocol stacks. Our platform supports IP mobility management, being able to use smooth network layer handoff techniques, including the optionally support of Fast Handoffs for Mobile IP (FMIP) [71, 72]. This protocol adds to the emulation platform an interesting feature to test those applications where seamless handoffs are vital to their successful deployment in vehicular environments with infrastructure domain.

In Chapter 4 we evaluate the QoS and Quality of Experience (QoE) of a video streaming service in a vehicular network using VESPA. Using VESPA we are able to evaluate QoE easier than using network simulators. The QoE concept considers much more than the performance of the network, in contrast with Quality of Service (QoS) evaluations. QoE is concerned with the overall experience the consumer has when accessing and using video streaming services.

A major portion of this chapter was published in [96].

Chapter Outline

The rest of this chapter is organized as follows: In Section 3.1 different vehicular network simulators are presented and are compared with VESPA. In Section 3.3, we present the design of VESPA. In Section 3.4 we analyze the accuracy and the scalability of the proposed platform. Section 3.5 concludes the chapter.

3.1 Vehicular Network Simulators

Two main types of simulators should be considered in vehicular networks simulation: network simulation and traffic simulation. To study vehicular networks, a simulator must be able to simulate not only network protocols but also vehicular movements. Regarding network simulators, they are usually used to test the functions and evaluate the performance of network protocols and applications under various network conditions. On the other hand, traffic simulators are usually used to simulate drivers' driving behavior (e.g., car following, lane changing, overtaking, etc.) on different kinds of vehicular scenarios (e.g., freeways, urban areas, etc.). Traffic simulators are usually used in the research areas of transportation engineering, such as transportation planning or traffic engineering.

It is important to use a realistic mobility model to obtain simulation results that correctly reflect the real-world performance of a vehicular network. For example, a vehicle node is typically constrained to streets which are separated by buildings, trees or other objects. In traffic simulations, four classes of traffic flow models are distinguished according to the detail level of the simulation: macroscopic, microscopic, mesoscopic and sub-microscopic models. In macroscopic models traffic flow is the basic entity. Microscopic models simulate the movement of every single vehicle on the street, mostly assuming that the behavior of the vehicle depends on both, the physical capabilities of the vehicle to move and the driver's ability to control it. Mesoscopic simulations are located at the boundary between microscopic and macroscopic simulations. Sub-microscopic models are more focused on single vehicles, like microscopic, but they extend the concept of vehicle by dividing it into further substructures, which describe the engine's rotation speed in relation to the vehicle's speed or the driver's preferred gear switching actions, for instance. Most traffic simulators for vehicular networks are built using microscopic models because it is the most appropriate model in order to get a good recreation of the performance of a vehicular network.

To the best of our knowledge, there are some integrated frameworks available for vehicular network testing. We can make two kinds of classifications of existing vehicular simulators.

In a first instance, we can classify vehicular network simulators as federated solutions or integrated solutions. Federated solutions are middleware software deployments to couple existing network and traffic simulators. Federated solutions usually provide a GUI to easily perform simulations using the capabilities provided by the network and traffic simulator. Federated solutions have the advantage of re-using existing (proven) software with good performance, avoiding the development of new software for the same functionalities. The conceptual architecture of a federated traffic/network simulator is shown in Figure 3.1.

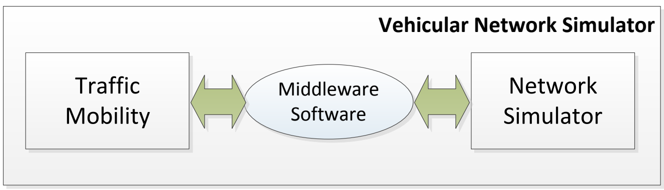

Fig. 3.1 The conceptual architecture of a federated traffic/network simulator

Three different methods are possible for constructing a simulator using the integrated approach and they are shown in Figure 3.2. In Figure 3.2a, communication model and network protocol simulation capabilities are added into an existing traffic simulator. In contrast, in Figure 3.2b, an existing network simulator is extended to include the capabilities of road network simulation and vehicle mobility models. Yet another method is to develop

(a) Network protocols added into a traffic simulator; (b) Mobility models added into a network simulator (c) New simulator developing all required components

Fig. 3.2 Three different methods for constructing an integrated traffic/network simulator

all required components from scratch to construct a new simulator, which is represented by Figure 3.2c. This kind of approaches has the advantage that traffic and network simulators are tightly integrated as a single program, so the feedback they provide to the other subsystem is very efficient.

Vehicular network simulators can also be classified into network-centric simulators and application-centric simulators. We show the network-centric and application-centric architectures of vehicular simulators in Figures 3.3 and 3.4.

The main component of the network-centric approach is the parser, which resides between the road traffic simulator and the network simulator. The traffic simulator generates a road network map and file that contains mobility-related information about all vehicles. The parser converts the mobility file into a mobility traces file, in a format acceptable by the network simulator. These mobility traces are fed to a network simulator as static input files.

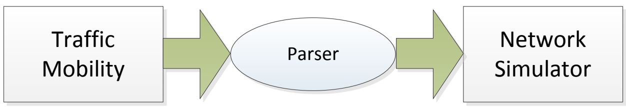

Fig. 3.3 Network-centric architecture

The application-centric approach allows the network simulator to control the mobility of vehicular nodes in simulation runtime. It is possible that vehicular drivers' behavior can change in reaction to vehicular safety applications. Therefore, in that case, it is needed to modify the mobility of selected vehicles, depending on the simulated scenario. Application-centric approaches give a feedback between the vehicle behavior and the mobility model and permit an evaluation of vehicular applications that influence vehicle's mobility. This feature is suitable for safety and traffic efficiency applications. For example, when a safety application broadcasts information reporting an accident, some of the neighboring vehicles may slow down. However, this feature is not so useful in the case of infotainment applications, like Internet access, multiplayer games, multimedia applications etc. It would be unrealistic

Table 3.1 Vehicular network simulators comparison

VANET simulator	Approach	Network Simulator	Traffic Simulator
TraNS	Federated/Application-centric	ns-2	SUMO
Veins	Federated/Application-centric	OMNet++	SUMO
SWANS/STRAW	Integrated/Network-centric	Jist/SWANS	STRAW
NCTUns	Integrated/Network-centric	NCTUns	NCTUns
iTETRIS	Federated/Application-centric	ns-3	SUMO
VNS	Federated/Application-centric	ns-3 or OMNet++	DIVER
TWINE	-	TWINE	-
VESPA	Federated/Network-centric	ns-2	SUMO

Table 3.2 Vehicular network simulators comparison

VANET simulator	Mobility	Emulation	IP mobility	IP micro-mobility	802.11p	Infrastructure capabilities	Oriented to
TraNS	Microscopic	No	No	No	Yes	No	Safety apps
Veins	Microscopic	No	No	No	Yes	No	Safety apps
SWANS/STRAW	Macroscopic	No	No	No	No	No	Safety apps
NCTUns	Microscopic	Yes	Yes	No	Yes	No	Both
iTETRIS	Microscopic	No	No	No	Yes	Yes	Both
VNS	Microscopic	No	No	No	Yes	Yes	Safety apps
TWINE	Microscopic	Yes	No	No	No	Yes	Infotainment
VESPA	Microscopic	Yes	Yes	Yes	Yes	Yes	Infotainment

to think that the vehicle's driver will reduce the speed due to the bad quality of the video. This is why VESPA does not implement this feature.

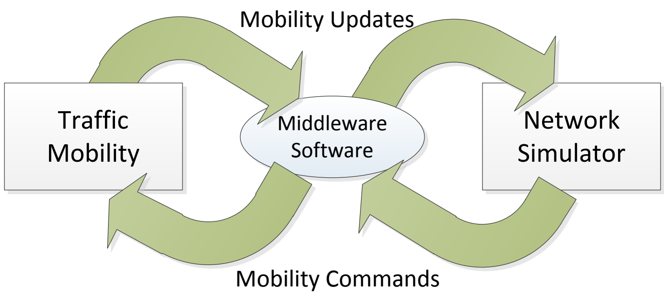

Fig. 3.4 Application-centric architecture

To provide access to a running road traffic simulation, application-centric approaches provides a specific interface for interlinking road traffic and networking simulators. In the existing approaches this interface uses a TCP based client/server architecture called Traffic Control Interface (TraCI).

In Table 3.1 and 3.2 we enumerate some vehicular network simulators that we are going to briefly survey and compare with VESPA. The table summarizes some features of these works. These works have been specially designed for research, and at least offer the functionalities of network and traffic simulation (as VESPA does). The simulators compared are not distributed under commercial licenses, because is a major impediment for their adoption by the research community. Below, we discuss these simulators.

The first presented solution is a federated vehicular simulator approach that links the traffic simulator SUMO [36] and the network simulator ns-2 [9] called TraNS (Traffic and Network Simulation Environment) [89]. TraNS is an open-source simulation environment that integrates both a mobility generator (SUMO) and a network simulator (ns-2) and it provides a tool to build realistic vehicular network simulations. TraNS is an application-centric approach and uses the specific interface for interlinking road traffic and networking simulators TraCI. TraNS features also includes support for realistic 802.11p [48]. Another interesting feature is the automated generation of road networks from the US Census Bureau Topologically Integrated Geographic Encoding and Referencing system (TIGER) [14] and Shapefile maps, and automated generation of random vehicle routes. However, it does not support the emulation feature, as VESPA does, and therefore it is not possible to test real applications with it. Also, TraNS is not capable of virtualizing nodes using real operating systems. In contrast with VESPA, TraNS is oriented to test V2V communications and does not support IP micro-mobility protocols.

VeiNS (Vehicles in Network Simulation) [17] is another open-source simulator that couples a mobility simulator with a network simulator using a federated approach. In VeiNS, SUMO is paired with OMNet++ [11] by extending SUMO to allow it to communicate with OMNet++ through a TCP connection. VeiNS, as TraNS does, allows the adaptation of drivers' behavior during simulation runtime to the vehicular network events using TraCI. VeiNS provides interesting features for safety applications, but the lack of support for network infrastructure and for network emulation features makes VeiNS not suitable to test real infotainment applications.

The Scalable Wireless Ad Hoc Network Simulator (SWANS) [33] is a Java based discrete-event network simulator that can be used as a network-centric vehicular network simulator, integrating the Street Random Waypoint (STRAW) [42] mobility simulator. SWANS gives the user the flexibility to build a custom application and execute it at the application layer, but it is not able to test real applications in real time. SWANS also lacks support for IP mobility protocols. The mobility simulator STRAW, provides to SWANS accurate simulation results by using a vehicular mobility model on real US cities, based on the operation of real vehicular traffic. STRAW is able to parse TIGER files, and it also implements a complex intersection management using traffic lights and traffic signs. However, its dependence on SWANS prevents the research community from using it. STRAW use a macroscopic mobility model that constrains node movement to streets defined by map data and limits their mobility according to vehicular congestion and simplified traffic control mechanisms. A more realistic mobility model with the appropriate level of detail for vehicular networks is critical for accurate network simulation.

NCTUns [109] is a discrete-event network simulator based on ns-2 that provides a complete GUI tool to configure testing scenarios easily. NCTUns can be used as a vehicular network simulator using its car agent. Therefore NCTUns is an integrated network-centric approach. NCTUns also supports emulation features. Despite the advantages provided by NCTUns, like the easiness of use, it presents several drawbacks. One of these drawbacks is that it only supports a predetermined fixed number of vehicles per simulation. NCTUns provides some random speed models, which are considered less realistic than the ones provided by SUMO. In addition, NCTUns requires Fedora 9 Linux distribution to be installed, limiting its usage. Finally, NCTUns is not as scalable as other platforms, as we will demonstrate later using some simulations (see Figure 3.10).

Another vehicular approach is iTETRIS [73]. The iTETRIS platform consists of SUMO, ns-3 network simulator and an Application module. iTETRIS is a federated and application-centric approach. All these blocks are connected by the iCS (iTETRIS Control System) module. Applications can be independently implemented and run on the top of the iCS using the Applications block. Triggered by Applications' commands, ns-3 simulates vehicular transmissions. Receptions deriving from these communications are notified to the applications, which in consequence can produce actions to be undertaken in the road traffic scenario simulated by SUMO using TraCI. As a result, SUMO continuously feeds the other blocks with vehicles' position updates, whose knowledge is essential for wireless simulations. iTETRIS permits synchronizing simulation time with the application, traffic or wireless communications events. However a real implementation cannot be run seamlessly over iTETRIS because this platform uses its own API to create network sockets. Therefore real applications cannot be tested using a real operating system, as opposed to VESPA. Moreover ns-3 does not support IP mobility, micro-mobility, Mobile IP or FMIP protocols. This is why iTETRIS is not suitable for analyzing infotainment applications in which handoffs and mobility issues are important.

The last vehicular approach presented is the Vehicular Network Simulator (VNS) [50]. VNS is a federated simulation framework that integrates either ns-3 or OMNet++ with DIVERT [49], a new microscopic traffic simulator. VNS provides bi-directionally interaction between the microscopic mobility model and network simulators, NS-3 and OMNET++, being an application-centric approach. VNS does not support mobility protocols and it is oriented to test safety applications.

The previous presented approaches are vehicular simulators. In the literature, there are other tools that can run real applications on modeled networks. In this respect, TWINE [112] emulator is one of the most used. TWINE targets realistic, scalable, and flexible evaluation of wireless technologies and applications. TWINE uses a geographically distributed set of physical wireless testbeds but this makes results difficult to be replicated by other researchers. TWINE is oriented to test wireless networks (wireless local area networks, mesh networks, or mobile ad hoc networks) but is not oriented to test vehicular networks. Thus TWINE cannot use realistic mobility models needed to evaluate vehicular applications. Moreover, to test real applications TWINE uses real devices instead of virtualized nodes, which is an important drawback in terms of scalability.

As a final remark, we can conclude that VESPA offers some extra features that current vehicular simulators do not offer. First of all, VESPA is a network-centric and federated approach. VESPA differs from the tools presented in the fact that it can work in emulation mode. Using this emulation feature and virtualization, VESPA is able to test real software in real time, injecting live traffic to the emulated vehicular network. Also, as the objective of VESPA is testing infotainment applications, it is essential to offer capabilities to communicate with the infrastructure side. Some of the alternative vehicular simulators do not offer this possibility of interacting with the infrastructure, maybe because they are focused on testing safety and traffic efficiency applications. Finally, to the best of our knowledge, VESPA is the only vehicular simulator that offers the possibility to test the effects of network mobility over real applications, and it also includes IP micro mobility protocols

3.2 Tools

Three main types of tools should be considered to emulate real applications in vehicular networks, as depicted in Figure 3.5: virtualized nodes, network simulation and vehicular traffic simulation. The virtualized nodes can be obtained by means of virtual machines, which are used to execute real applications without necessity of a broad number of hardware resources. Network simulators are used to evaluate network protocols in a variety of conditions. Traffic simulators are used for transportation and traffic engineering. In the rest of this Section we summarize the different tools used to construct VESPA.

3.2.1 User Mode Linux - UML

Working with virtual machines can be cost saving because management can be simplified, as there is just a unique point to control, in contrast to managing and setting up a lot of dispersed machines in the network. This is especially important in vehicular networks because deploying a fleet of cars is extremely costly and hard to manage. In our particular case, we will use UML machines hosted in a single machine to virtualize applications. UML provides open source Linux virtual machines, fast speed and good performance, and it can

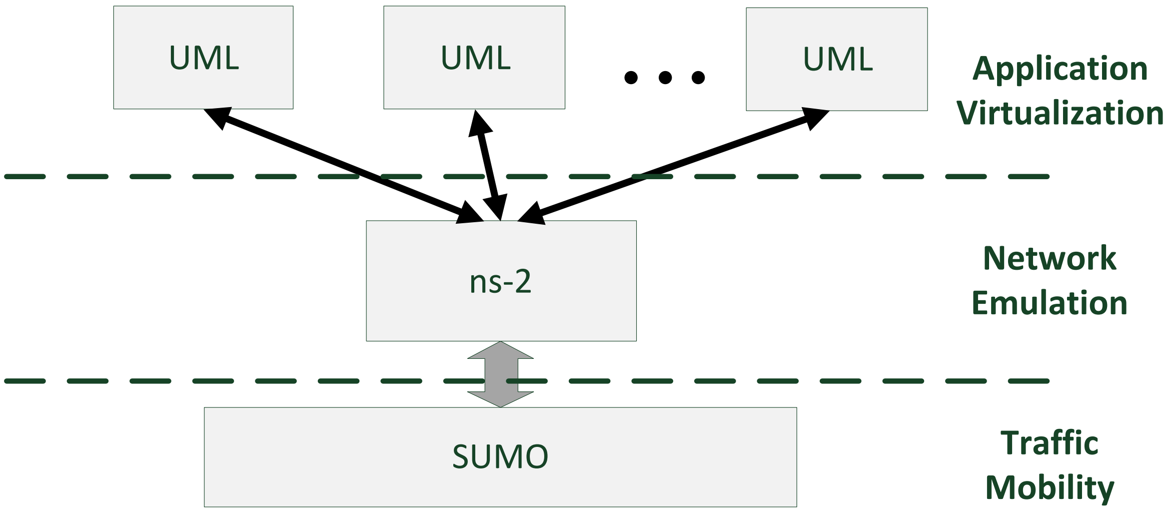

Fig. 3.5 VESPA modules

be easily managed by using a single file for the kernel of the virtual machine and another file that contains the file system which will be used. UML is lighter and may work better than others virtualization systems because there is less instruction translation involved, it just intercepts the system calls and throw them back to the UML kernel [47]. At present, our platform is based on a single host machine, so the number of vehicular nodes that can be represented using virtual machines is limited, and it mainly depends on the available resources (CPU, memory, storage, etc) of this host machine. We are considering as a future work to distribute virtualization to avoid this limitation and to extend its scalability.

3.2.2 Network Simulator - ns-2

A network simulator is a software program that mimics the working of a computer network. Simulators typically model the computer network with devices and links of any type. They also offer support for the most popular protocols in use today, including medium access control (MAC), routing and transport protocols, and some simple applications. Hence, network simulators allow us to analyze the network applications performance and to test new protocols in the data link, network or transport layers. Some network simulators also have an emulation facility, that is to say, the ability to introduce the simulator into a live network. In our particular case, as our objective was to develop a framework able to test real applications in real time, this emulation facility was mandatory. For this and many other features (support for network mobility protocols, GPL license, etc.) we finally decided to use ns-2 as network emulator in our testing framework VESPA. We also considered ns-3 [10] as a candidate for the network emulator, because it has some usability and performance advantages regarding its predecessor ns-2. However, ns-3 currently does not have full support for network mobility protocols, for instance Mobile IP, which is essential for properly testing vehicular networks.

3.2.3 NS-2 Emulation Extensions

The NS-2 Emulation Extensions [79] are part of the contributed code of ns-2. This means that they are maintained by users and that they have not been incorporated into the regular ns distributions. It is used to link ns-2 simulator with UML virtual machines, enabling ns-2 to emulate wireless networks using real software. In these extensions, the scheduler of the network simulator has been enhanced for the correct emulation of wireless networks, solving some timing inaccuracies that produce a negative impact over the performance of the IEEE 802.11 protocol in ns-2 [78].

3.2.4 SUMO

One of the most important parameters when simulating vehicular networks is node mobility. It is important to use a realistic mobility model to obtain simulation results that correctly reflect the real-world performance of a vehicular network. Traffic simulators provide realistic mobility traces to network simulators. Network simulators use these mobility traces to calculate the network conditions of vehicular nodes, performing channel modelling as a function of geoposition. In our case, we will use SUMO [36] as traffic simulator because is an open source road traffic simulation package designed to handle large road networks that can be easily integrated with ns-2 simulator. SUMO can import many network formats and combining with OpenStreetMaps (OSM) [12], so we can easily simulate vehicular traffic mobility using any map imported from the Internet. Since SUMO is a pure traffic generator, its generated traces cannot be directly used by the available network simulators. However SUMO also is able to generate alternative XML traces, which can be easily converted to the format of ns-2 mobility traces.

3.3 VESPA modules

VESPA consists of a set of software developments and a GUI tool that integrates three basic modules: a node virtualization module based on UML virtual machines, a network simulator module based on ns-2 network simulator, and a vehicular traffic mobility module based on SUMO. But VESPA is more than the grouping of these existing modules. We have used some different software add-ons and we developed a set of tools for the proper interconnection between them. These add-ons permit to connect the simulator to the UML virtual machines, to optimize the network emulation for working in real-time with wireless devices and to have support for the IEEE 802.11p/DSRC vehicular technology. These add-ons also permit to offer a set of features that are required to emulate a vehicular network including an infrastructure domain. We have developed a simplistic GUI tool for VESPA, which facilitates the use of some complex tools such as ns-2 and SUMO. The rest of this Section describes in more detail the logic modules that are part of VESPA, which are depicted in Figure 3.6.

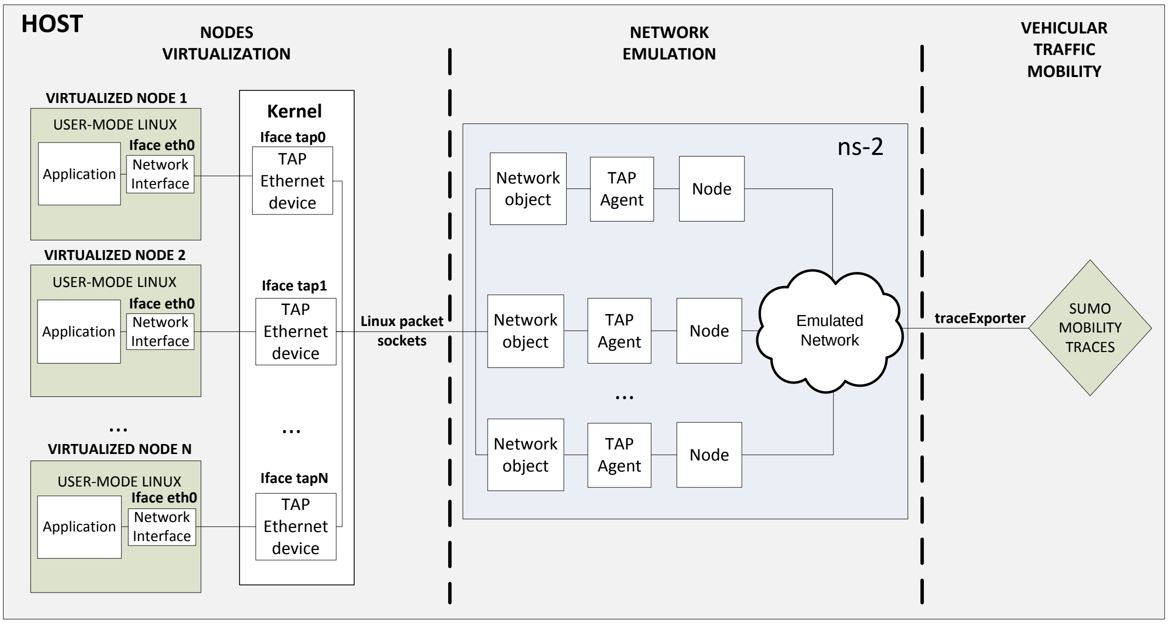

Fig. 3.6 VESPA modules

3.3.1 Nodes Virtualization

As it can be seen in Figure 3.6, the top modules consist of nodes virtualized by UML machines, representing several nodes in the network setting up a live network.

Each filesystem of a UML virtualized node is entirely contained inside a single file on the host. VESPA allows the use of shared filesystems between several virtual machines using the copy-on-write (COW) layering capability. Each virtual machine saves the changes in the filesystem into a COW file, much smaller than the original filesystem, without modifying the original filesystem file. This leads to a disk space saving. It will also help performance, since the host will be able to cache the shared data using a much smaller amount of memory, so UML disk requests will be served from the host's memory rather than its disks.

The applications that are intended to be tested using VESPA are installed in these virtual machines.

As stated in Section 3.2.1, node virtualization is expensive in terms of resources (CPU, memory, storage, etc.), so there is a limitation in the number of guest machines that can be virtualized in a particular host. However, just notice that maybe it is not necessary to virtualize all the nodes that exist in the vehicular network, but only those in which we want to test real software. VESPA allows that some nodes can be virtualized with UML machines with the aim of executing the real applications, and other nodes can be just modeled within the ns-2 network simulator. For instance, in the test scenario that we will be presented in Chapter 4, it is only necessary to virtualize two nodes as UML virtual machines, the video streaming server and the vehicular video client. The rest of nodes involved in the reference test scenario may be just emulated inside the ns-2.

3.3.2 Network Emulation

As previously stated, UML virtual machines can represent both mobile vehicles located in the wireless domain or fixed nodes (usually servers) located in the infrastructure domain. These virtualized nodes need connectivity, which is provided by the ns-2 network simulator. To connect the virtualized nodes with ns-2, VESPA uses TAP virtual Ethernet devices [13] to transport information to and from UML virtual machines (see Figure 3.6). A TAP device is assigned to each UML, which is used to connect the virtual node to the emulated network. In particular, UML machines see these devices as a common ethernet device that is directly connected to the corresponding virtual Ethernet interfaces.

To emulate the network, VESPA uses the emulation feature of ns-2 (nse). Nse uses a soft real-time scheduler which ties event execution within the simulator to real time. Ns-2 acts as an emulator of a wireless network among the virtual machines. This emulation feature mainly uses *network objects* and *tap agents*. NS-2 Emulation Extensions provide the network objects and the tap agents used in VESPA. The network objects are used to send and receive packets to and from a live network. Network objects read and write packets to tap network devices at the link layer. Packet sockets available in Linux are used for this purpose. The tap agents are application level processes on ns-2 nodes that convert network packets between the emulated wireless network and the real network using a network object to access to a network device on the link layer (TAP virtual interfaces). Each tap agent can be connected to at most one network object. Tap agents additionally implement address mapping between UML virtual machines, MAC addresses and the ns-2 nodes addressing.

The main problem we found in this software was that the original tap agents available in NS-2 Emulation Extensions were initially developed to test wireless scenarios without infrastructure. However, most infotainment applications depend on infrastructure, so there was a lack in this aspect. For this reason, we modified the existing tap agents to allow the interaction with the infrastructure domain and to properly interact with network mobility protocols. We modified the implementation of the NS-2 Emulation Extensions because they are not totally compatible with the ns-2 mobile IP implementation.

In addition to these changes, some extensions were needed to enhance the network emulator for a better performance when representing vehicular networks. Next we describe these add-ons.

Mobile IP in ns-2

The ns-2 network emulator supports Mobile IP for wired and wireless networks. Mobile IP is used to track the location of any mobile terminal in order to deliver any packets to it whenever its location and it is available on ns-2 to simulate mobility scenarios. The Mobile IP ns-2 module was developed by Sun Microsystems. It includes all Mobile IP entities defined in Mobile IPv4 [84], like Home Agents (HA), Foreign Agents (FA) and Mobile Nodes (MN). HA and FA entities are deployed as access points (APs), and these entities have registering agents to send beacons to the MN to detect mobility, to encapsulate and decapsulate data, and to reply to solicitations from MN. The MN has registering agents, which receive and respond to beacons and send out solicitations to HA or FAs.

As VESPA has been designed to be especially suitable for testing real-time applications, we consider essential to provide smooth handoff techniques, such as FMIP (Fast Handoffs for Mobile IP), apart from the standard Mobile IP implementation. FMIP uses cross-layer techniques to “anticipate” or to prepare for the forthcoming handoff beforehand, minimizing the handoff latency and packet losses. Using FMIP, VESPA is able to emulate scenarios where smooth handoffs are necessary for an acceptable application performance. To provide this FMIP support, an extension developed by Robert Hsieh [3] has been added to ns-2.

Routing

The five different ad-hoc routing protocols currently implemented for mobile networking in ns-2 are DSDV [37], DSR [67], AODV [85], TORA [15] and PUMA [108]. VESPA is able to use them to emulate Vehicle-to-Vehicle (V2V) applications. However, these ad-hoc routing protocols do not properly consider mechanisms for efficient mobility management and handoff support. Then, there can be some incompatibilities between the ns-2 implementation of these routing protocols and the implementation of Mobile IP and FMIP. Infrastructure is needed to provide IP mobility, which typically is assumed to be a fixed and wired backbone composed of routers and access points to provide mobility services to wireless terminals. Only DSDV can work in ns-2 with infrastructure and IP mobility at the same time, and at the expense of not using multi-hop techniques. However, the use of a proactive protocol like DSDV in a vehicular scenario can introduce too much traffic overhead. This is why we have configured VESPA to use the NO Ad-Hoc Routing (NOAH) agent [8] by default. When using the NOAH implementation, VESPA emulates the behavior of mobile nodes that communicate without using ad hoc routing, so the mobiles nodes are only able to connect with the access points. This behavior is suitable to test infotainment applications that are provided by the infrastructure.

IEEE 802.11 extensions

The team from Mercedes-Benz Research & Development North America (MB) and from University of Karlsruhe has collaborated to develop a completely new 802.11 MAC and PHY model for ns-2, called Mac802_11Ext and WirelessPhyExt, respectively [41]. This new model allows to configure a lot of new parameters of the MAC and PHY layer that are not possible to configure in the current ns-2 implementation, providing a higher level of simulation accuracy. Using these extensions, it is possible to use the IEEE 802.11p access technology in ns-2, using a configuration file that provides the 802.11p parameters. For that reason, we have included this extension in VESPA.

3.3.3 Vehicular Traffic Mobility

Finally, regarding the last module of traffic simulation, we decided to use SUMO in VESPA. Some of the reasons that helped us to choose SUMO were that it is the most used traffic simulator licensed under GPL, it has a quite good documentation, and it is easy to interconnect with other software. SUMO offers a good number of features to build mobility patterns in vehicular networks, like the possibility to create maps by means of theoretical models (e.g. Manhattan grid), or to import real maps from external sources. SUMO also permits to configure aspects related to nodes movement, like the number of vehicles, maximum speed, etc.

SUMO generates *netstate* dumps that contain information about the nodes' position and speed. These dumps are generated once we have loaded a mobility scenario, and they have to be converted to suitable mobility traces. A parser module named *traceExporter* converts *netstate* dumps to ns-2 mobility traces. If so, ns-2 can use them as an input in order to calculate the network conditions of these nodes. Traces are calculated once a user introduces a SUMO configuration file to VESPA, or after a user has built a SUMO scenario with the

Graphical User Interface (GUI) provided in the emulation platform, capable to edit traffic scenarios.

3.3.4 Graphical User Interface

A simple and intuitive GUI (see Figure 3.7) has been developed to facilitate the use of VESPA. This intuitive GUI makes testing trouble-free and efficient, in contrast to program-driven systems which require complex programming or scripting. VESPA's graphical software lets you zoom in on low-level parameters without having to go through manuals or specifications. The GUI's clean interface makes it easy to "dive deep" and control the fine details of emulating a complete network. This GUI helps in editing mobility scenarios and running the tests, as it provides a set of utilities that automatically create the configuration files needed for the emulation.

Fig. 3.7 The emulation platform GUI

The GUI help us: (1) to create random maps, (including grid maps, spider networks and totally random maps) and vehicular random routes; (2) to create maps manually, using an editor where the map nodes and the edges between these nodes can be created. When a manual map is created the routes must be edited manually too. This is done using another utility where different routes can be created and then assigned to different vehicles; and (3) to import external maps and vehicular routes. Once the map and the routes configuration files are created, VESPA offers the possibility to create a SUMO configuration file. This configuration file can be used to load this mobility scenario a posteriori in the emulation platform. Regarding other network parameters, it is possible to configure the virtual nodes, the RSUs and the network emulation duration. The virtual nodes must be configured, selecting the nodes that must be virtualized using UML virtual machines. The other existing nodes will be directly emulated by ns-2. Regarding the RSUs, it is possible to configure the position coordinates and the access technology parameters (IEEE 802.11a, 802.11b or 802.11p) of each one. The parameters used for each technology are provided by [41]. All RSUs deployed in the map will be connected to a central server that will be virtualized by a UML virtual machine.

3.4 VESPA's Accuracy and Scalability

We ran a set of tests to gauge the performance limitations of our emulation platform. In particular, we assess the impact on the accuracy of the emulation as it inevitably introduces delay when network traffic is sent between applications running on the virtual machines and the real-time simulator. All these evaluation tests have been performed on a Quad-core (1.6 GHZ) Intel x64 system running Debian-Linux with kernel version 3.2.0-35. The host machine has 32GB memory, the UML machines use at most 32MB of RAM, and the logging process of the simulator uses 100MB for the compressed trace file. Inside the UML machines we used the same 2.6.31 Linux kernel as on the host system.

To evaluate VESPA's accuracy, we compare the measurements obtained via emulation with the results of pure ns-2 simulation. To determine the round-trip times (RTTs), we have used simple ping (ICMP echo) measurements. We evaluated the delays introduced by the emulated network, including virtualization and the traffic redirection, and those introduced by the simulation model. We execute the "ping" command to send 10,000 ICMP packets from one virtualized node to another. The results are shown in Figure 3.8[1] as error bars under various payload sizes. This clearly indicates that the RTT increases for packets with a bigger size, which can be explained that it takes longer to fully place a big packet on the

[1] We increased the kernel's default interrupt frequency such that 1 jiffy becomes 1 ms in the modified kernel, i.e., so that the packet delay experiment results' accuracy is within 1 ms

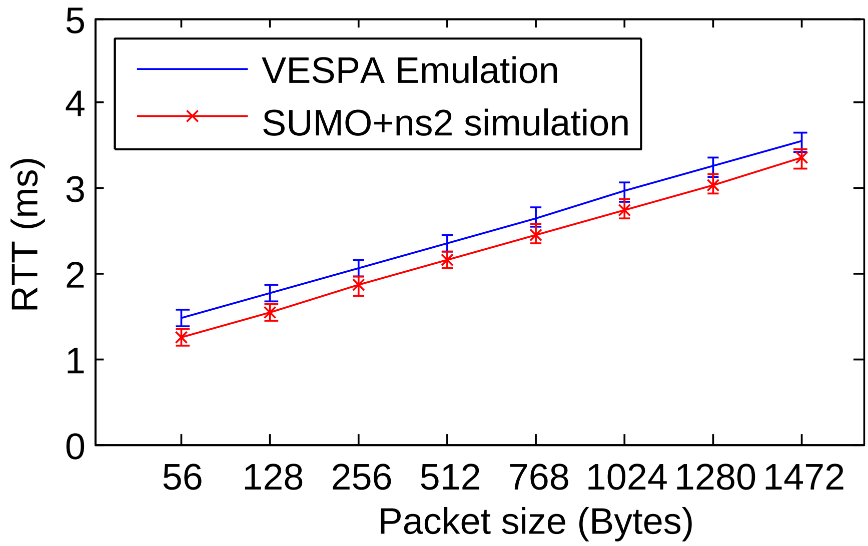

Fig. 3.8 RTT with various payload sizes

medium. It can be also seen that the emulation results correspond accurately with the simulation results, with exception of a latency overhead of about 0.15 ms in the emulation case due to the additional packet handling layer in the virtual machine. Note that the standard deviation of the RTT values in VESPA is comparable to the measured standard deviation of the simulation results.

To evaluate VESPA's scalability, we measured the resource requirements in terms of CPU utilization and memory requirements as the number of emulated nodes is increased. The load is generated using the following scenario: vehicles are moving at constant speed, they are separated 5 meters from each other and they communicate with 802.11p. Each node opens a socket and sends fixed-sized UDP packets to the a server in the wired domain at a constant rate. In order to characterize the overhead we vary the traffic rate from 50 to 10K packets per second (pkt/s), using packet sizes of 100 and 1000 bytes. We use `vzmemcheck` command to get the memory consumption for the UMLs. For CPU load, we used the `vmstat` command. As shown in Figure 3.9, the memory consumption is linearly increasing with the number of virtual nodes. This is because that each virtual machine is a separate executing entity, with constant memory occupancy. On the other hand, the CPU utilization reaches 90% with around 20 virtual nodes due to the decrease in the instructions per communicated byte when increasing the number of emulated entities. In Figure 3.10 we can see the ratio of late packets in terms of packet size. To that end, we virtualized 4 nodes and we used the same traffic pattern that in the other experiments. Compared to TWINE, VESPA maintains a lower ratio of late packets and therefore it is able to maintain the real-time during the tests better than TWINE.

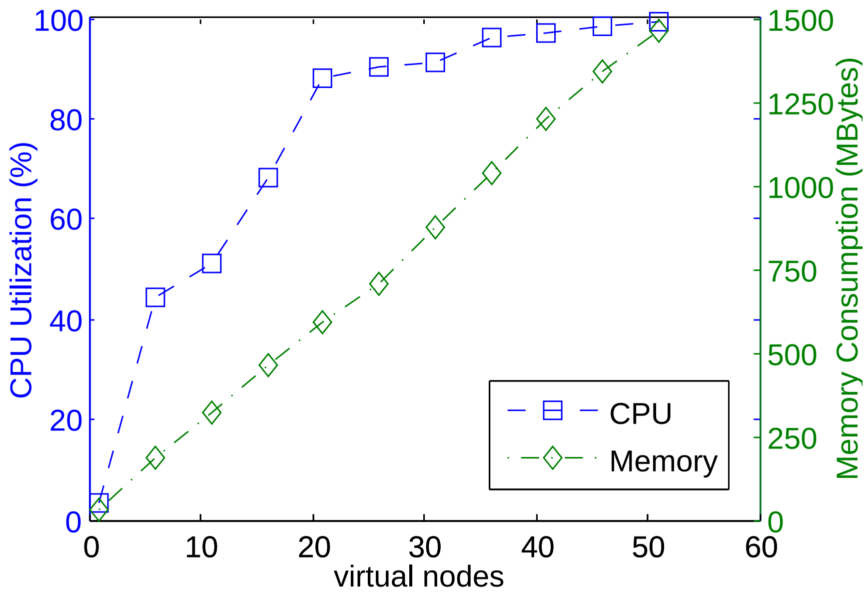

Fig. 3.9 CPU and Memory Utilization

3.5 Conclusions

Numerous vehicular simulators have been proposed in the literature to analyze the performance of vehicular networks. In this chapter, we have presented and analyzed different simulation approaches oriented to test wireless applications in modeled vehicular scenarios. Unfortunately, these simulators are oriented to test safety applications based on V2V communications, and do not permit to test real applications. Thus it is needed to generate simplified versions of the applications to test, being a waste of time, and making it difficult to test infotainment applications with real-time requirements.

VESPA emulation platform allows testing real infotainment applications using vehicular traffic mobility and providing a faster and cheaper testing environment than outdoor experiments. Researchers can use this low-cost tool to test and analyze the deployability of different infotainment vehicular services in a realistic scenario. VESPA is mainly aimed at researchers and developers that want to test applications in a vehicular scenario in a very easy way. VESPA can be easily installed in Linux systems or downloaded as a Live CD, being able to be executed in any desktop computer. VESPA can be used, for example, to compare the performance of various codification techniques (or video players) in a controlled vehicular scenario, or to compare different video codification techniques. With VESPA, it is possible to test applications using the same software developed for desktop computers without lasting time in modeling these applications for network simulators, avoiding the limitations caused by simplified application behaviors. Moreover developers can test their

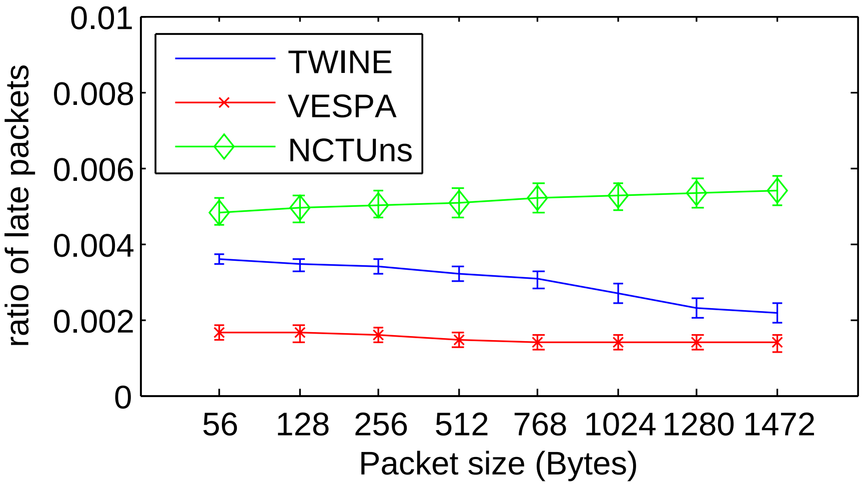

Fig. 3.10 Ratio of late packets

software in a complex mobile scenario in a straightforward and fast manner.

Chapter 4

Performance Evaluation of Multimedia Infotainment Services Using VESPA

Video applications and video content are expected to be a growing infotainment service, such as real-time traffic information broadcasting or various on-road video entertainments (live sports, news, etc.). Video streaming applications are resource-consuming and they have some constraints that should be considered (and tested) before starting the deployment of the infrastructure necessary to provide them. Regarding resources, vehicles can be equipped with powerful CPUs, and large *on-board* memory and storage capacities. The IEEE 802.11 standard can support transmission rates higher than 100 Mbps, and specifically the vehicular amendment IEEE 802.11p [48] standard support up to 27 Mbps. Even between high speed driving vehicles within a highway, it is reasonable to expect a 1Mbps data rate [102]. The car engine and battery can provide power for intensive data computation, communications and to feed the video displays. Thus, vehicles can be considered powerful enough to play video flows, and the network have enough bandwidth to support the transmission data rate required to send, receive or forward compressed video flows among vehicles and roadside receivers. But the mobile Internet was not designed with video requirements in mind and therefore video infotainment applications may be handled inefficiently in vehicular networks. Vehicular network handoffs still face a challenging difficulty: the high mobility of vehicles creates frequent handoffs, which may result in significant packet delay and packet losses. As a consequence, the deployability of a video service over such environment must be analyzed. Despite the importance of reliable results, nearly all ongoing research activities addressing video streaming over vehicular networks are based on V2V communication simulation studies [66, 103] that neglect the effects of frequent handoffs over real video applications.

In this chapter we present a performance evaluation for video infotainment applications with infrastructure participation in vehicular networks. We present a study for the potential deployment of video on demand services in vehicular networks where a Mobile IP solution is used for real-time video using UDP+RTP protocols. We carried out a set of experiments using a video streaming service in a simple scenario (a highway). The video server was placed in the infrastructure domain and vehicular nodes download contents from this server during a trip. In our experiments we only considered the effect of losses due to handoffs, without any other traffic interference. So, we will evaluate the effects of handoffs between access points caused by the mobility of vehicles.

In a real scenario, handoffs can be performed at link layer (L2) or at network layer (L3). L2 handoffs can be used when all the RSUs belong to the same subnet and to the same administrative domain. L3 handoffs are most often found in wired and wireless environments where users need to carry their mobile devices across multiple LAN subnets. This behavior fits in a vehicular environment where a vehicular node is moving through different RSUs placed along a road. So we decided to simulate the worst case, in which handoffs are performed at layer 3 and a network-layer mobility protocol is used in all of them. We simulated network-layer handoffs using Mobile IP and Fast Handoffs for Mobile IP (FMIP) protocols [71] and we analyzed the performance of a video playback in a highway scenario with both protocols. In such simplistic scenario, we will only evaluate the effects of packet losses caused by handoffs due to network mobility. There are other problems that can limit the deployment of a video service, for example, packet loss during handoffs, random errors in wireless links, network congestion, etc. However, the performed analysis is focused on the packet loss that occurs during the handoffs due to network mobility. Trying to test more parameters to the analysis may degrade the performance of the simulations. According to our simulations, we will extract some interesting results about the deployability of such service.

For this work we used the testing framework presented in Chapter 3 called VESPA. Using VESPA we are able to evaluate the Quality of Experience (QoE) easier than using network simulators. The QoE concept considers much more than the performance of the network, in contrast with Quality of Service (QoS) evaluations. QoE is concerned with the overall experience the consumer has when accessing and using video streaming services. As previously stated, VESPA has the possibility to install and use real software. In particular, we will use the Live555 [51] libraries for the multimedia applications. This fact will also allow us to compare the video stream transmitted by the server with the one received in the video player. This feature of VESPA, that other existing vehicular simulation tools do not have, permits us to identify losses during handoffs in a realistic way. Analyzing the simulation results we noticed that, even in this “friendly” scenario, there are problems to deploy properly a video service in highways because its performance at high speeds is quite poor.

A major portion of this chapter was published in [95].

Chapter Outline

The rest of this chapter is organized as follows: In Section 4.1, we present the reference scenario used for the performance evaluation. In Section 4.2, we present the results of the performance evaluation Section 3.5 concludes the chapter with a discussion of the results obtained.

4.1 Reference Scenario

As previously stated, we will use VESPA to analyze the effects of (only) losses caused by L3 handoffs. The test scenario designed for this purpose is an infrastructure scenario where a set of RSUs are deployed over a highway in an overlapped manner (see Figure 4.1). Therefore, there are not coverage blackouts in the road. All the RSUs are connected to a central router, which is also connected to a video streaming server. Both the video streaming server at the infrastructure side and the vehicular node with the video player are emulated by UML [46] virtual machines. The Live555 [51] libraries provide the multimedia applications for the test bed. Using these libraries, a video streaming server is configured in the infrastructure side, and a video receiver is placed in the vehicular node. Live555 is an open source library that can be used to build multimedia streaming applications, and that provides different tools for testing purposes. It supports multimedia transport and application open standards such as RTP/RTCP, RTSP or SIP. Live555 supports video and audio formats such as MPEG, H.264 and JPEG video, and it has been designed in such a way that it can be easily extended to support more formats. Using these libraries, a video streaming server is configured in the infrastructure side, and a VLC media player [18] or an MPlayer player[7] with live555 libraries is placed in the OBU of the vehicular node. During our tests we found several problems because the video player crashes because it was not able to reproduce the video stream in case there are many errors. For this reason, we developed an error-resilient decoder that does not crash and offers a video reproduction without stops even in the presence of errors or gaps to the user. Our decoder is based on the MPEG-2 decoder [6], and it recovers the gaps that are lost during the communication by representing the previous frame received when a lost frame is detected.

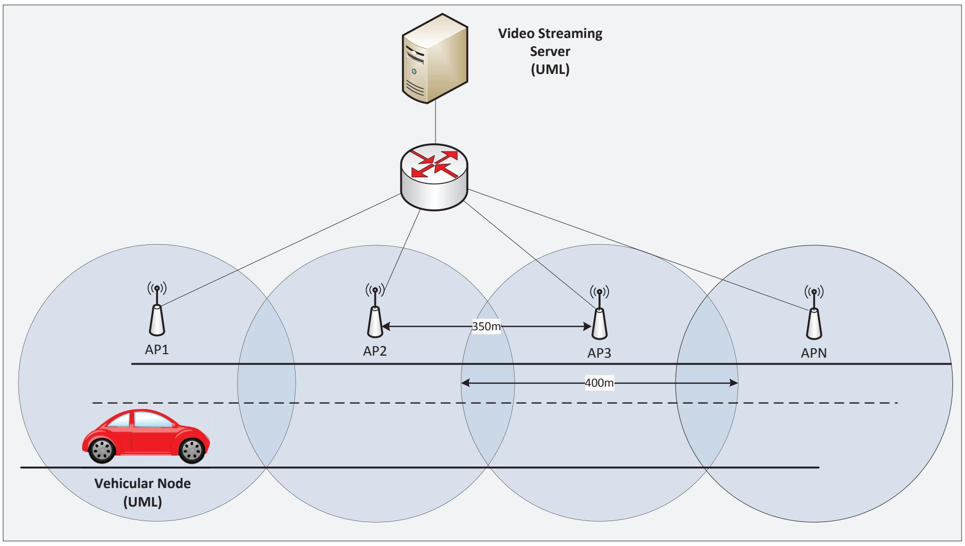

Fig. 4.1 Reference scenario

The video transmitted during the simulations have a CIF (Common Intermediate Format) format (352x288). CIF, also known as FCIF (Full Common Intermediate Format), is a format used to standardize the horizontal and vertical resolutions in pixels of YCbCr sequences in video signals, commonly used in video teleconferencing systems. The video streaming media is sent using the User Datagram Protocol (UDP). There is no mechanism within UDP to guarantee delivery. It is up to the receiving application to detect loss or corruption and to recover data using error correction techniques. If data are lost, the stream may suffer a dropout and therefore quality degradation. The Real-time Streaming Protocol (RTSP), Real-time Transport Protocol (RTP) and the Real-time Transport Control Protocol (RTCP) are used in conjunction with UDP, and they were specifically designed to stream media over networks. There are other protocols that are reliable, such as the Transmission Control Protocol (TCP), which guarantee correct delivery of each bit in the media stream. However, TCP reliability is based on a system of timeouts and retries, which means that when there is a segment loss in the network, the media stream stalls while the protocol handlers detect this loss and retransmit the missing data. Clients can minimize this effect by buffering data. While delay due to buffering is acceptable in some video services like video on demand scenarios, users of interactive applications (such as video conferencing) will experience a loss of fidelity if the delay caused by this buffering exceeds 200 ms. This is why we simulated video streaming services using RTP+UDP protocols in a lossy scenario with several handoffs.

Regarding routing issues, we will assume during the test the simplistic routing case in which there is only one hop between vehicular nodes and RSUs, so there are no multi-hop communications.

Parameter Name	Value
Wired links	Bandwidth: 100Mb Propagation delay: 5ms
Propagation model	Nakagami
Wireless access	IEEE 802.11p
Distance between RSUs	350m
Video characteristics	CIF 352x288 MPEG-2

Table 4.1 Simulation parameters

Some other parameters used during the simulations are described in Table 4.1. The distance between consecutive RSUs is 350 meters. The node radio coverage is 400 meters, therefore a vehicular node is always under the coverage of an AP. The IEEE 802.11p access technology is used in the simulations. The propagation model used is the Nagakami model with the parameters used in [41] by default. We choose this propagation model because empirical research studies have shown that a fading radio propagation model, such as the Nakagami model, is best for simulation of a vehicular environment [106].

4.2 Simulation Results

4.3 Packet Loss Rate

In this test we sent several video streams coded at different bitrates using CBR (Constant Bit Rate) coding. Our objective was to analyze and compare the effects of the standard Mobile IP handoff and FMIP over the packet loss rate (for further information about the working of these protocols see Appendix A). Mobile IP model implemented in ns-2 used in VESPA follows Mobile IP for IPv4 standard. Also, we wanted to gauge the effects of mobility over the video transmission. We streamed a video in this reference scenario, during 4200 meters along the road using different vehicle speeds. Therefore the simulation time is variable as a function of the vehicle speed, to assure that the number of handoffs during the video playback is always the same for the different speeds.

Figures 4.2 and 4.3 show the packet loss rate as a function of the bitrate and the vehicle speed when using Mobile IP. We simulated speeds between 20 and 50 m/s. Although the speed limit in highways is usually between 30 and 40 m/s we considered interesting simulating until 50 m/s because there exists highways where there is no speed limit. From these figures it can be observed that, using Mobile IP, the packet loss rate increases as vehicle speed increases, but when the CBR video bitrate increases there is not a considerable increase in the packet loss rate. This behavior is due to the disruption time for higher speeds is higher than disruption times for slower speeds. When increasing vehicle speeds disruption time also increase, however when increasing bitrates disruption time remains.

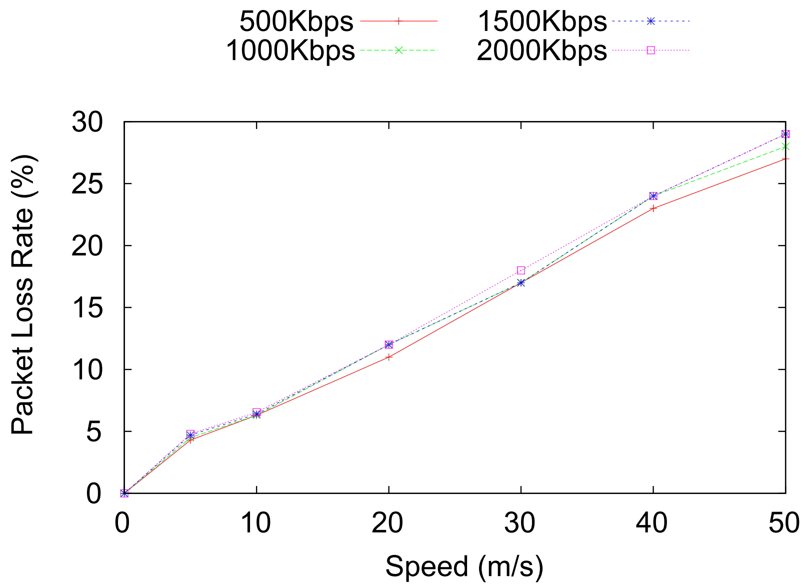

Fig. 4.2 Packet loss rate per vehicle speed using Mobile IP

From Figures 4.2 and 4.3 it can be concluded that the values of the packet loss rate obtained when using the Mobile IP solution can be a problem to deploy a video service in a vehicular network. The values obtained during the simulations are comprised between 4% and 29%. The packet loss rate obtained at 5 m/s is between 4% and 5% and the obtained at 50 m/s is between 27% and 29%. Driving at those speeds typically achieved in a highway (30-40 m/s) the packet loss rate is between 17% and 24%.

Next we will test the behavior of the FMIP protocol, because this protocol is supposed to offer seamless communications. FMIP can reduce the handoff delay by either introducing L2 triggers to anticipate the handoff, or by managing most of the handoff operations inside a local domain. Minimizing the handoff delay, the FMIP standard reduces the amount of lost packets during the L3 handoff. However, FMIP does not always guarantee a successful predictive fast handoff if the speed of the mobile node is high[1]. The handoff process of FMIP tightly depends on L2 triggering, and it can increase the possibility of failure because the trigger does not consider the state of the L3 of the mobile node and it delivers triggers only based on variable wireless signal state. So, although in FMIP the packets are buffered

[1] Differences between predictive and reactive FMIP handoffs are detailed in Appendix A.

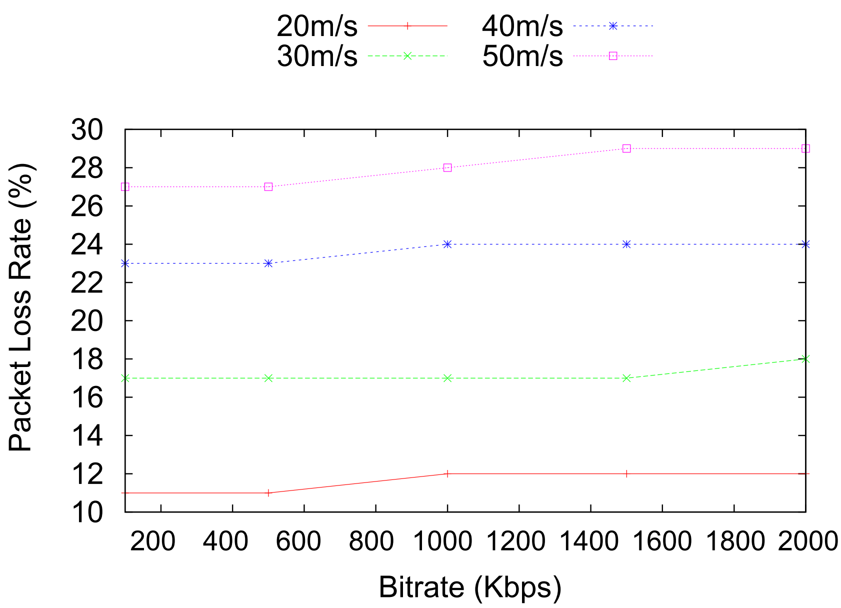

Fig. 4.3 Packet loss rate per bitrate using Mobile IP

and supplied to the mobile node after the handoffs to avoid packet loss, unsynchronized L2 triggers with the L3 status will generate a reactive handoff that can produce packet losses close to MIP handoffs that affect to the video streaming services.

Figures 4.4 and 4.5 analyze the packet loss rate using the FMIP protocol as a function of the bitrate and the vehicle speed. From the figures it can be concluded that the values of the packet loss rate obtained when using the FMIP solution are lower than the simulations using Mobile IP. Also, the behavior of the packet loss rate is different. From these figures it can be observed that, using FMIP, the packet loss rate increases as the vehicle speed increases, and it also increases with the increase of the CBR video bitrate. This behavior is explained because the number of unsuccessful FMIP handoffs depends on these factors. The increment of unsuccessful FMIP handoffs is caused by the loss of messages in the FMIP handoff anticipation mechanisms (see Appendix A). These failures consist in the achievement of the handoff between subnets before the configuration of the new network parameters and before the tunnel establishment to forward the packets on the fly. These failures cause packet losses and happen more often with the increase of the node velocity and the increase of the bitrate. According to Figures 4.4 and 4.5, the packet loss rate is comprised between 1.5% and 28%, but being much lower than Mobile IP at slow and medium speeds. For instance, driving at speeds typically achieved in a highway (30-40 m/s), the packet loss rate is between 4.75% and 21%. For example, going at 30m/s a 1 Mbps video will suffer a packet loss rate lower than a 15% when using FMIP techniques.

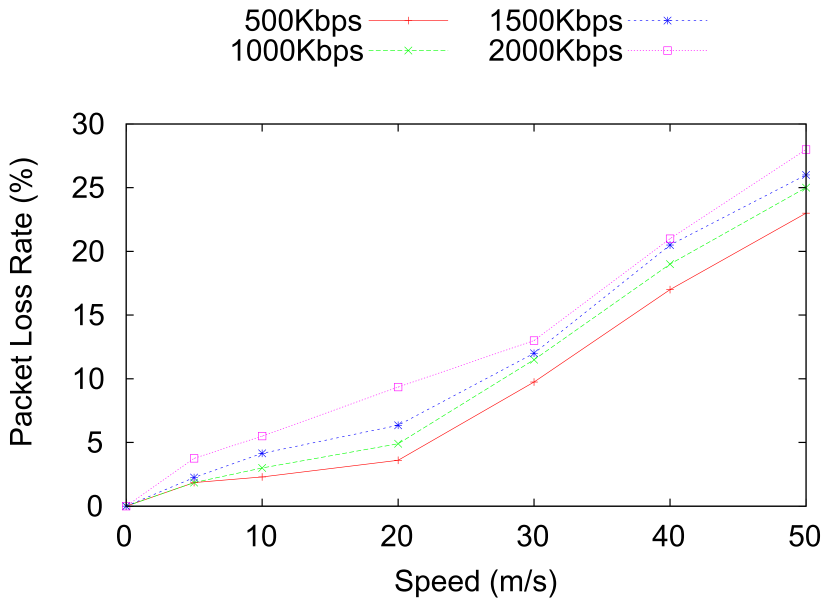

Fig. 4.4 Packet loss rate per vehicle speed using FMIP

4.3.1 Video quality

The objective of these simulations is to see how the quality of a video clip streamed in a vehicular network is affected by the handoffs occurred during the communication. During these handoffs, packet losses limit the overall quality of the video streamed because the transport protocols used during the transmission (UDP or UDP+RTP) are not reliable. In contrast to the simulations in Subsection 4.3, in the analysis detailed below we can see the impact of the packet loss distribution during the handoffs in the quality of the video transmitted.

A major feature of video encoding is the ability to remove redundant information, not only within a frame, but also among a group of frames. MPEG-2 uses three frame types (Intra-coded, Predicted and Bi-predictive) to represent the video. A group of pictures (GOP) setting defines the pattern of the three frame types used. These three picture types are defined in the following ways. I-frames are the most important in all three types of frames, since the decoder uses the content in I-frames to decode the P-frames and B-frames in the same GOP. If the content is lost in the I-frame, the error will be propagated to all the other frames in the same GOP. A packet loss in a P-frame or B-frame only affects to this particular frame or at most to some few neighbor pictures.

QoS metrics, such as packet loss rate, have been consistently used for evaluating the quality of a transmission in IP networks and provide an objective way to measure the reliability of communication network. However, Quality of Experience (QoE) metrics address

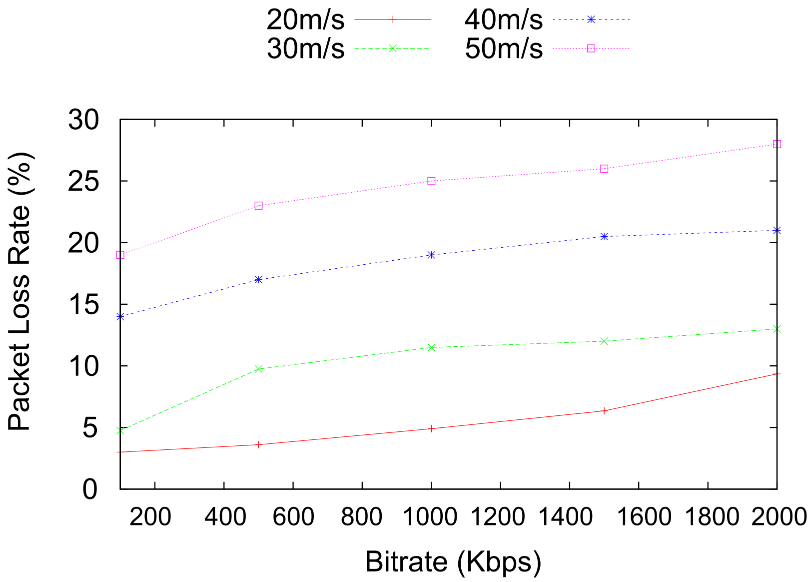

Fig. 4.5 Packet loss rate per bitrate using FMIP

the limitations of conventional QoS measuring when evaluating quality from the user's point of view. Frame loss derived from packet loss suffered in vehicular networks can affect the video in different ways depending on whether the packet lost belongs to a slice from one or another frame type. So we need different metrics to measure what a user perceives as a quality parameter. From a technical point of view, QoE can be seen as the quality remaining in the user's device after delivering the video to an end device. PSNR is considered an objective QoE metric [28].

So apart from typical QoS, such as packet loss rate, we analyzed the quality of the video streamed using the Peak Signal-to-Noise Ratio (PSNR) between the original video and the video transmitted within the vehicular network and the one received by the vehicular node. That means the distortion introduced to the video until it reaches the decoder at the end device. PSNR is defined via the mean squared error (MSE) which for two $m \cdot n$ images I and K where one of the images is considered a noisy approximation of the other.

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2 \tag{4.1}$$

The MSE is computed using (4.1) where i and j represent the coordinate of a pixel along the horizontal axis and the vertical axis respectively, and where m and n represent the height and width of the video sequence. As such $I(i,j)$, denotes a pixel value in the original video sequence at coordinate (i,j), where the video sequence in question has a spatial resolution of $m \cdot n$. Similarly, the notation denotes a pixel value in the received video $K(i, j)$ sequence, having a spatial resolution of $m \cdot n$.

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) = 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \tag{4.2}$$

The PSNR values are calculated using (4.2) where MAX_I defines the maximum luma value (MAX_I is equal to 2^8 when the pixel depth is equal to eight bits per pixel component). Note that the PSNR values are computed in the luma domain (Y-PSNR), since the luminance component is the most widely accepted objective measure of visual distortion and therefore is our primary means of measuring visual distortion [70].

PSNR is most commonly used as a measure of quality of reconstruction of lossy compression codecs. So, in order to be able to compare the same content, we developed an error-resilient decoder based on MPEG-2 [6] that recovers the lost gaps of lost frames during the communication by representing the previous frame received.

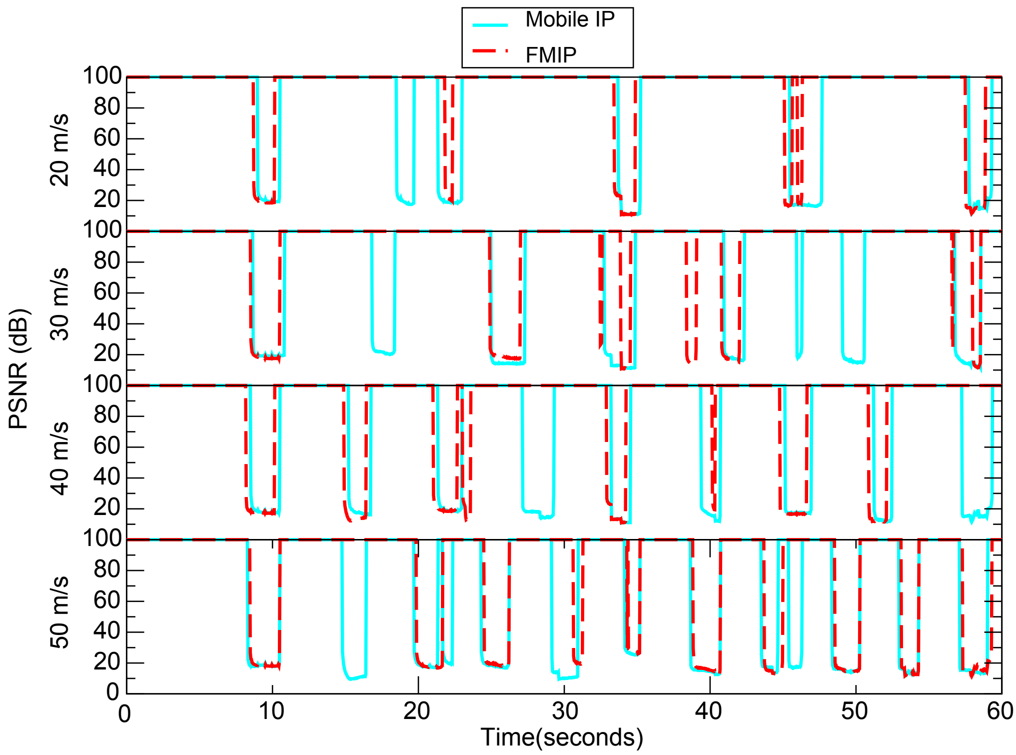

Fig. 4.6 PSNR of a CBR 1000Kbps video for different vehicle speeds

Figure 4.6 shows the PSNR for four different speeds between 20 and 50 m/s using a video encoded at CBR with 1000 Kbps. Here the video playback time is always the same for all the vehicle speeds. We can observe the video degradation during handoffs using

Mobile IP or FMIP. When no handoffs occur and therefore there are no losses during the video transmission, the PSNR value is infinite because the compared videos are the same (MSE=0), but we will represent this infinite value as 100 dB in the Figure.

It is generally accepted that a PSNR value between 30 and 50 dB represents a good quality level. However we can observe that during handoffs the PSNR obtained is lower than 20 dB. The PSNR shows an on-off behavior (between 100 and 20 approximately) produced by the lost frames during the handoff disruptions. So we can consider that during handoffs the quality level is not enough, obviously because there are missing frames and the user perceives stops during the playback. It is easily observed that, increasing the speed of the vehicular node, the PSNR is degraded due to the increase of the number of handoffs during the communication. We can also observe the difference in the PSNR degradation during the handoffs between Mobile IP and FMIP. At low speeds, FMIP seamless handoffs produce that just some of the handoffs cause degradation on the video streamed. However when the vehicular node speed is increased the benefits from using FMIP in the video playback are diminished. For the sake of example, at 50 m/s, it is difficult to see differences between Mobile IP and FMIP protocol in the video quality perceived in a video streamed. We can observe that at lower speeds some FMIP handoffs produces no losses or shorter blackouts than Mobile IP handoffs. When no blackout exists during a FMIP handoff means that it is a predictive handoff and the video frames transmitted to the vehicular node are tunneled correctly to the next AP during the handoff without losses. When a blackout exists during a FMIP handoff means that it is a reactive handoff. A reactive handoff occurs when the L2 handoff and therefore the disconnection is previous to the Fast Binding Update (see Appendix A). A blackout in the video playback is always produced by Mobile IP handoffs because there is no *make-before-break* technique between the new link detection and the information transfer period and therefore the handoff causes packet loss.

Figures 4.7 and 4.8 represent the average PSNR (accumulated error) after 200 seconds of video. Using the Average PSNR we can observe the video degradation for different bitrates as a function of the vehicular speed. It can be observed that the average PSNR is decreased as the vehicle speed is increased. This can also be deduced from Figure 4.6 since it can be easily observed that handoff disruptions are more present at high speeds. In the same way, it can also be observed that the average PSNR is lower with higher bitrates. Figures 4.7 and 4.8 represent the degradation of the video played caused by the mobile scenario, so video degradation caused by packet loss during handoffs is more significant with the increase of the video bitrate.

Figures 4.9 and 4.10 represent the video disruption time during the playback for Mobile IP and FMIP handoffs. Table 4.2 shows the video disruptions information in numerical

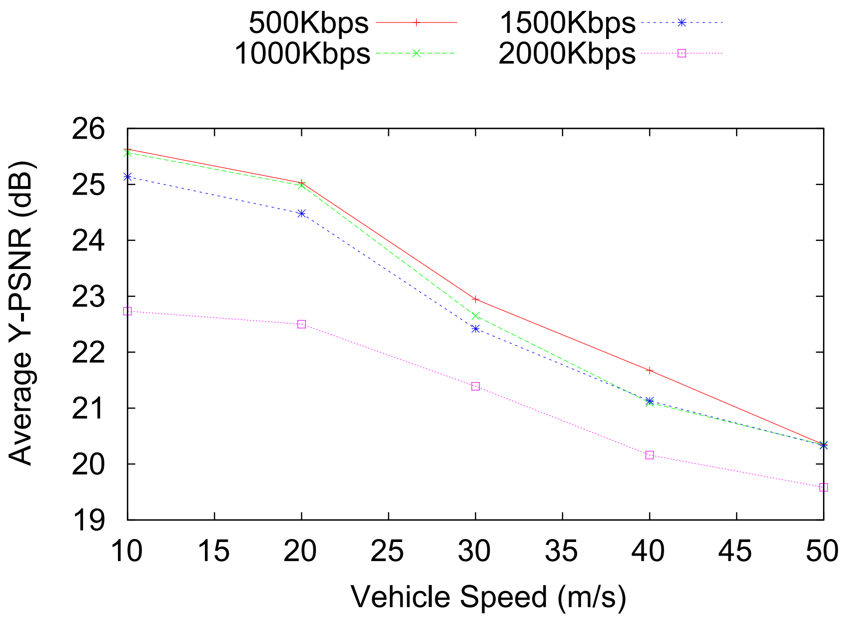

Fig. 4.7 Average PSNR per vehicle speed using Mobile IP

	Mobile IP				FMIP			
Speed	20m/s	30m/s	40m/s	50m/s	20m/s	30m/s	40m/s	50m/s
Number of video disruptions	19	25	33	39	11	23	25	35
Disruption Mean (sec)	1.6337	1.6064	1.6545	1.7692	1.0982	1.3496	1.5	1.5497
Disruption Variance	0.2703	0.235	0.2372	0.3096	0.6011	0.9039	0.2183	0.1885
Number Lost Frames	796	1004	1365	1725	302	706	939	1356
% Lost Frames	15.48	19.53	26.55	33.55	5.87	13.73	18.26	26.37

Table 4.2 Video Disruptions

values. The video playback observed is about 200 seconds long. It can be observed that the number of disruptions and the disruption time increases at higher speeds because the number of reactive handoffs also increases. At lower speeds, predictive FMIP handoffs are predominant and therefore the disruptions perceived in contrast with Mobile IP are lower. The variance for FMIP disruption time is higher for lower speeds since the behavior of the reactive handoff is more random than the behavior of a simple Mobile IP handoff. However this variance for higher speeds is similar between both since the behavior of the handoffs at high speeds are very similar (in FMIP L2 handoffs are suffered always before any handoff message due to the high speed). According to [110], the FMIP protocol can reduce the handoff delay to get between 0.18 and 0.4 seconds in 99.3% of the cases. However, the analytical model for fast handoff latency presented in [110] does not take into account the node speed, and it is oriented to predictive handoffs. Therefore, for reactive fast handoffs the latency will be close to Mobile IP handoff latency. Predictive FMIP handoffs cannot be perceived during the video playback due to buffering techniques. Anyway, our results are

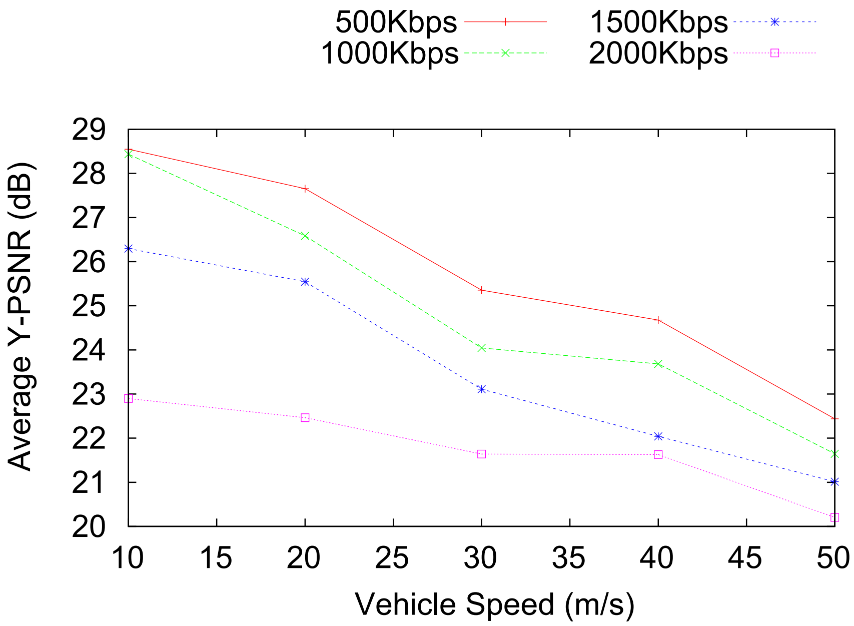

Fig. 4.8 Average PSNR per vehicle speed using FMIP

very close to the ones presented in [61] for reactive FMIP handoffs.

According to our results, it seems that it is not possible to offer this kind of service with enough quality for the users using UDP+RTP. The video playback degradation is unacceptable in several cases, especially at high vehicle speeds.

4.4 Conclusions

Analyzing the results obtained in the performance evaluation, we will try to extract some remarks about the possibility of deploying a video service in a highway with current techniques. A potential business target for video streaming services is high-range vehicles. These vehicular users are the potential customers that are willing to pay for video services in car trips, and probably these customers will not accept degradations of the video service due to high speeds. QoS should be maintained even in this case. This issue can be more important in countries where the law does not limit the overall speed at certain highways. Just recall that we considered a simplistic scenario in which there are no coverage blackouts, there is no multi-hop routing and all communications are with the infrastructure. In addition, during the simulations we only considered the effects of packet losses during L3 handoffs, and no other important aspects (like transmission errors, losses due to congestion, bandwidth variations due to weather effects, etc.) were considered that would worsen the

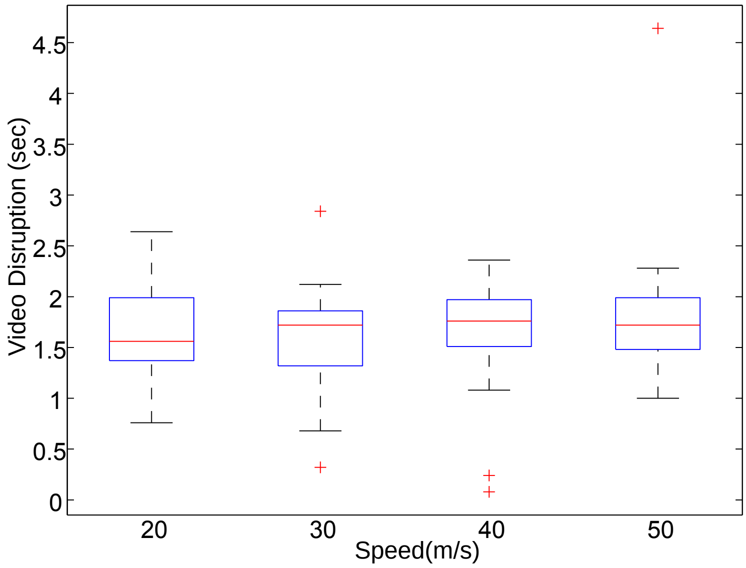

Fig. 4.9 Video Disruption per vehicle speed using Mobile IP

performance of the system for sure. According to our results, it seems that it is not possible to offer this kind of service in all the scenarios or with enough quality for the users. The packet loss rate is unacceptable in several cases, specially at high vehicle speeds.

For this reason, we think existing techniques are not practical to minimize blackouts communications during handoffs to deploy video streaming services in highways. In our opinion, one of the reasons that deter the deployability of such kind of service is the use of non-reliable protocols to transport video services, for instance UDP or UDP+RTP. As these protocols do not recover lost frames, losses during the handoffs cause gaps in the video. The use of reliable transport protocols, like TCP, would allow recovering gaps due to lost frames. However, reliable protocols like TCP generally add delays to request lost frames. So, reliable protocols are not suitable for real-time services, such as video conference, but they can be suitable for video on demand services. In this case, we can use prefetching techniques to avoid the video from freezing during the playback caused by lost frames. Buffering should not be a problem for vehicles since OBUs are not considered resource-limited devices. Using video prefetching, received frames are stored at the receiver buffer. After an initial prefetching time where the buffer is storing arriving frames, the receiver starts to play the media. The stored frames allow the video users to continue playback during handoffs, avoiding video freezing. In order to check the deployability of video services using TCP we simulated the same scenario than Section 4.1. In the simulation depicted in

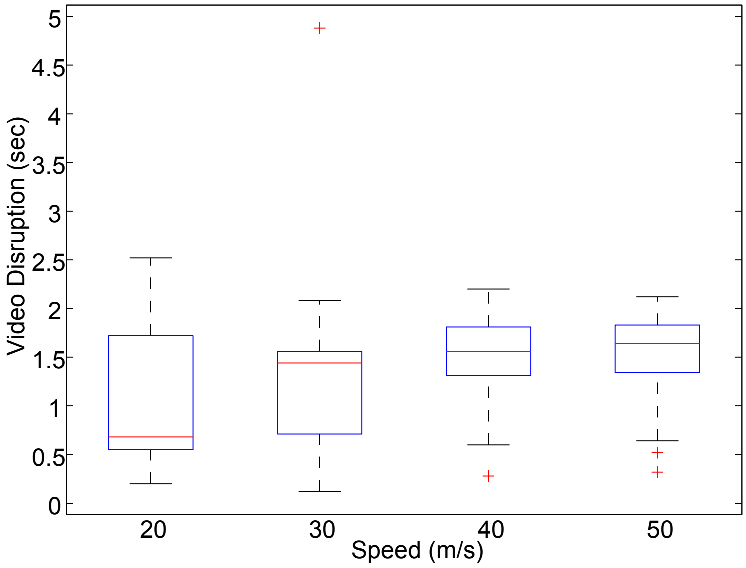

Fig. 4.10 Video Disruption per vehicle speed using FMIP

Figure 4.11 we represent the buffer occupation used for the video prefetching. We used a buffer of 600KBytes and the video playback starts after a prefetching of 300KBytes.

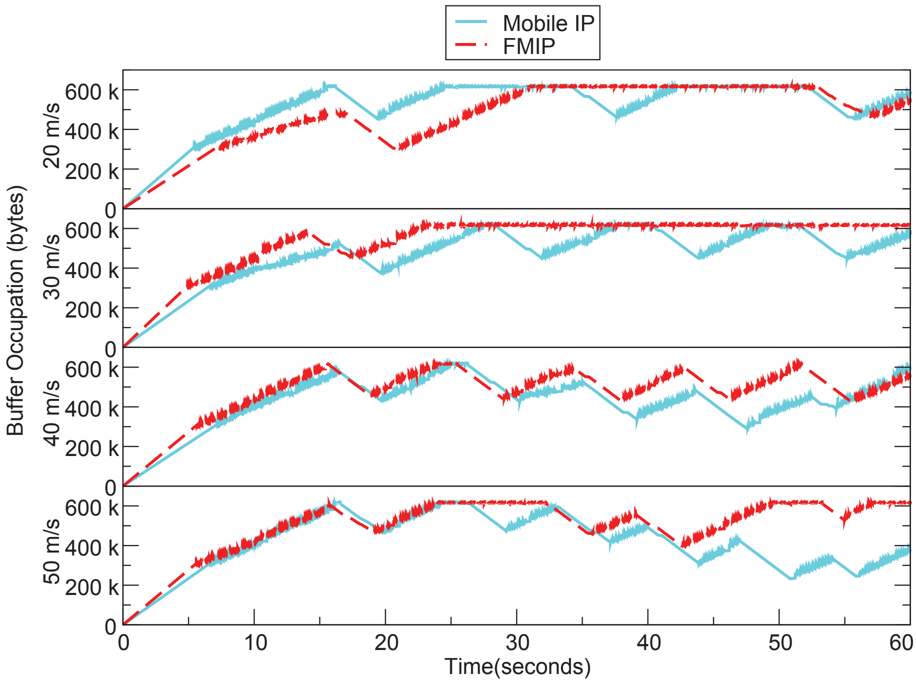

Fig. 4.11 Buffer occupation for a CBR 1000Kbps video transmitted using TCP for different vehicle speeds

The buffer has been sized following the results obtained in Figure 4.6. Since we can observe that the biggest gap in the communication is approximately 2.5 seconds, we need a buffer bigger than 300KBytes. We sized the buffer to the double of this value (600KBytes) to avoid the buffer underrun during handoffs and the video freezing caused by it. We can observe that buffer occupation is maintained, therefore there will not be video disruptions during the video playback. However, using Mobile IP for speeds higher than 50m/s, a forbidden speed in most of the highways, the buffer could be underrun since it can be seen that the high number of disruptions causes an alarming drop of the buffer occupation.

Another possibility to reduce losses during handoffs may be to use advanced video coding techniques, such as application-layer forward error correction (FEC) using rateless erasure coding combined with Scalable Video Coding (SVC) extension of H.264/MPEG4-AVC [29, 99].

Another possibility to reduce or eliminate losses during handoffs may be to use a reliable TCP connection, but avoiding delays and TCP performance issues (introduced in Chapter 2) using cross-layer techniques. This way, we could use video streaming applications using a smaller buffer, and therefore adding a smaller delay in the video playback. In next chapter, we present a new handoff architecture for V2I communications that, preventing from TCP impairments during handoffs, allows more seamless communications and an enhancement of TCP throughput rates and TCP fairness between different vehicular nodes.

Chapter 5

VSPLIT: A cross-layer architecture for V2I TCP services over 802.11

Some non-safety services on ITS rely on the Transport Control Protocol (TCP), one of the core protocols of the Internet Protocol Suite. However, there exists several issues related to mobility that can affect TCP performance, and these issues are particularly important in vehicular networks.

To achieve Internet communications in vehicular networks, handoff procedures perform data flow migrations from a Correspondent Node (CN) (e.g., an Internet server placed in the wired network) and a Vehicular Node (VN). In these handoffs the VN changes the Point of Attachment (PoA) in the infrastructured network, and these PoAs can have the same or different access network technologies. Therefore, handoffs can be intra-technology or inter-technology. Handoffs within PoAs of the same access network technology are intra-technology, also named horizontal handoffs. Handoffs within different access network technologies are inter-technology, also named vertical handoffs. This work is focused on horizontal handoffs, specifically on handoffs between different 802.11 RSUs that act as PoA, placed along the road. We considered IEEE 802.11 as access technology because it is the main candidate technology in vehicular networks due to its flexibility and cost, as detailed in Chapter 2.

Handoffs cause several problems to TCP communications. Communications are paused while the handoff between RSUs is completed, and packets can be routed again to the VN. In any case, TCP considers that any loss is a congestion signal, and reacts dropping its congestion window (cwnd), which causes a reduction in the transmission rate. TCP fairness is also important in vehicular networks. There can be unfairness after a handoff between the user that performs the handoff and other users still connected to this RSU, specially when vehicular nodes go at different speeds. Also, conventional TCP congestion-control algorithms cause an overshooting window problem and result in poor throughput when used in 802.11 networks [54]. MAC contention in 802.11 channels cause an increment of the Round-Trip Time (RTT). This behavior can overload the network because this RTT increase does not imply an increase in the capacity of the network.

In this chapter we present VSPLIT, a new handoff architecture for V2I communications to optimize the handoff procedure in TCP communications by using the IEEE 802.21 MIH services [21]. MIH services are a set of mechanisms mainly used to facilitate migration of mobile users between access networks that use different link-layer technologies (more information available in Chapter 2). However, we use these mechanisms to enhance TCP performance during handoffs in V2I communications. We use a TCP-splitting architecture for vehicular environments, in which a modified TCP protocol is used between a Performance-Enhancing Proxy (PEP) and the vehicular users. The presented architecture allows not only to reduce the handoff disruption time for TCP communications during handoffs, but also to increase the aggregated throughput of all the vehicular users in the network and to enhance the fairness between TCP connections. This modified TCP protocol is called VSPLIT-TCP, and it uses cross-layer information provided by MIH services to improve the performance of TCP flows in the vehicular segment. VSPLIT-TCP uses the standard TCP headers, the TCP standard flow control (based on a sliding window) and also the TCP standard error control (based on retransmissions and time-outs). However, VSPLIT-TCP does not implement the standard congestion control. Instead, VSPLIT-TCP uses cross-layer metrics provided by IEEE 802.21 MIH services to assist handoffs over 802.11, so VSPLIT-TCP can adapt its congestion window to the characteristics of the network condition of the next RSU. In particular, VSPLIT-TCP is able to synchronize the connections/disconnections, avoiding packet loss, timeouts or spurious retransmissions during a handoff. The TCP flow is frozen during the idle period when a handoff occurs. VSPLIT-TCP does not implement any probing phase (slow start or congestion avoidance). Instead, VSPLIT-TCP uses the available cross-layering information to properly set the congestion window of active senders according to the efficiency and fairness conditions at each moment. Therefore the congestion window is configured properly just after the VN is reconnected without having to wait any probing phase.

The work of this chapter was published in [97].

Chapter Outline

The rest of this chapter is organized as follows: Section 5.1 reports on the related work. In Section 5.2 we present the TCP handoff architecture proposed. In Section 5.3 the handoff procedures for CN-to-VN flows and for VN-to-CN flows are respectively detailed. The performance evaluation of the proposed architecture is presented in Section 5.4. Section 5.5 concludes the chapter.

5.1 TCP Handoffs Approaches

Several cross-layer schemes have been proposed in the literature to alleviate handoff issues when using TCP in wireless scenarios. Here we summarize some of the most important approaches and we include some key properties in Table 5.1.

Name	Handoff	Technology	Flow Direction	802.21	Cwnd Adaption	Required Modifications	Approach
3DA [40]	Horizontal	Generic	CN-to-MN	No	Halve	MN	End-to-End
Freeze TCP [58]	Horizontal	Generic	CN-to-MN	No	Freeze	MN	End-to-End
ATCP [101]	Horizontal	Generic	CN-to-MN - MN-to-CN	No	Freeze	MN	End-to-End
SWHA [76]	Vertical	WLAN and DVB-S	CN-to-MN and MN-to-CN	No	Yes	MN	End-to-End
Inter-RAT [75]	Vertical	WLAN and WiMAX	CN-to-MN	Yes	No	MN and Snooping Agent	Snooping
THAT [57]	Vertical	WLAN and DVB-S	CN-to-MN	Yes	Yes	MN	End-to-End
VSPLIT	Horizontal	WLAN	CN-to-MN and MN-to-CN	Yes	Yes	MN and PEP	Splitting

Table 5.1 TCP Handoffs Approaches Comparison

The *3DA* [40] is a horizontal handoff approach that is aimed to reduce the idle time after a handoff in a WLAN. This approach uses the AP beacons to control the handoff procedures of a Mobile-Node (MN). After a reconnection, the TCP receiver sends three copies of the last received Acknowledgement, triggering the TCP sender to enter the TCP fast-retransmit–fast-recovery (FR-FR) mechanism [104]. The solution is aimed to CN-to-MN TCP flows. The cwnd is not adapted to the new link and the FR-FR mechanism halves the cwnd after a handoff. This approach reduces the overall handoff latency but, after running FR-FR, the effective transmission rate is halved without adapting it to the new link characteristics.

In [58], a TCP variant named *Freeze-TCP* is proposed. *Freeze-TCP* forces an interruption of the TCP flow to avoid packet loss during handoffs by advertising a zero window (ZW). This allows stopping transmission without dropping the TCP cwnd. Afterward, the TCP sender sends a “probe” message to retrieve the transmission with the unchanged cwnd. This mechanism implies that the link layer properly notifies the handoff occurrence to the transport layer. *Freeze-TCP* is designed for horizontal handoffs, but it does not specify the access technology used nor the mobility protocol.

Adapted-TCP (ATCP) is a cross-layer TCP for wireless horizontal handoffs [101]. Unlike *3DA* and *Freeze-TCP*, *ATCP* improves the performance not only when the TCP sender is the CN but also when the TCP sender is the MN. *ATCP* runs only at the MN and, after a disconnection and a connection notification from the network layer, it appropriately handles internal TCP parameters to allow a fast restart of the TCP transfer after handoff. *ATCP* does not prevent packet or ack losses during handoffs and if the sending window is closed and the RTO is larger than the handoff, the *ATCP* sender must wait for a RTO expiration before restarting the TCP transfer. *ATCP* considers the state variables that are achieved before the disconnection as the optimum value also over the new link. In *3DA*, *Freeze-TCP* and *ATCP*, the authors assume Mobile IP protocol as a mobility solution.

In [76], the authors propose a scheme aimed to completely avoid losses during a handoff through a cross-layer architecture that optimizes the performance of TCP-based applications when a handoff occurs between links with quite different BDPs. This cross-layer architecture is focused on vertical handoffs between satellite and WLANs. To this scope, cross-layer signaling involving transport, network, and link layers has been designed. It is compliant to the ECLAIR architecture [93], an "optimization subsystem" (OSS) that manages cross-layer signaling, whereas "tuning layers" (TLs) interact with each protocol layer to drive optimized actions. We named this solution *Satellite-WLAN Handoff Architecture (SWHA)*. The cross-layer solution is implemented just at the MN and it is an end-to-end solution. SWHA can be used for CN-to-MN and MN-to-CN TCP flows. Using the cross-layer feedback the MN close the cwnd window to stop the transmission for MN-to-CN flows or sends a ZW for CN-to-MN. After the handoff the connection is restarted at full cwnd. However this solution does not specify how the optimal cwnd is calculated for each kind of handoff.

In [75] the authors provide a seamless handoff procedure between UMTS and WiMAX systems using MIH 802.21 standard. They design a new TCP agent (TCP Snoop), which interacts with a MIH Function variant called InterWorking (IW) layer, to mitigate BDP mismatch and to solve spurious RTO problems that often appear in the inter-technology handoff scenarios. We named this solution *Inter-RAT Handoff*. The TCP Snoop Agent reacts with a 3-dupack and a ZWA after a handoff indication to halve the cwnd of the TCP sender and freeze the transmission. The Snoop Agent uses the BDP advertised by the IW to dimension the Snoop Agent buffer. After a handoff completion a Nonzero Window Advertisement (NZWA) message is sent from the Snoop Agent to the TCP sender and with the Acknowledgement delaying mechanism and explicit window notification resolve BDP mismatch and spurious RTO problems. Inter-RAT just takes into account spurious RTO problems for TCP communications in WiMax to UMTS handoffs and CN-to-MN TCP flows.

In [57] the authors present a solution to provide a seamless service to high-speed trains. They propose a cross-layer architecture with vertical handoffs between a satellite link in open areas and WLAN in tunnels. We named this solution *TCP Handoff Arquitecture for high-speed Trains (THAT)*. They designed appropriate vertical handoffs with time anticipation for an uninterrupted service using 802.21 Satellite Independent - Service Access Point (SI-SAP) interface of the Broadband Satellite Multimedia standard. They propose different methods based on a reordering and RTO update approach and the configuration of the TCP slow-start threshold and the cwnd limitation on the basis of the AP buffer size for satellite-to-WLAN handoffs and the restoration of the previous slow-start threshold value in WLAN-to-satellite handoffs.

The solution proposed, named VSPLIT, is a TCP-splitting architecture for horizontal handoffs in vehicular environments. However it can be considered a solution between horizontal and vertical handoffs. Using the IEEE 802.21 MIH services typically used in heterogeneous networks, the TCP can be informed about service disruptions during handoffs and about the characteristics of the following 802.11 link to adapt its cwnd to the new situation. Notice that, despite handoffs are between the same access technology (horizontal), link characteristics can be different due to several reasons (number of users connected to each RSU, network load, mobility, etc.), situation that typically occurs in vertical handoffs. VSPLIT can be used for CN-to-MN handoffs (CN-to-VN in vehicular networks) and for MN-to-CN handoffs, and it requires MN modifications and a PEP between the CN and the MN. Despite VSPLIT is mainly aimed at 802.11 link-layer handoffs, it can also be applied to Mobile IP handoffs.

5.2 VSPLIT: TCP Handoff Architecture

In this section we introduce VSPLIT, an approach that helps to solve some mobility issues in the TCP protocol for V2I communications in vehicular networks that use 802.11 as an access network technology. To counteract impairments at the TCP protocol that are caused by handoffs we propose an approach that is based in a TCP splitting architecture. Then, using splitting, the connection is divided in two TCP segments: standard TCP segment and vehicular TCP segment (see Figure 5.1). The proposed architecture uses a modified TCP in the vehicular TCP segment named VSPLIT-TCP, which minimizes the packet loss in TCP connections during the handoff process, adapts TCP to the characteristics of the next RSU of the network and provides fairness among vehicular users. VSPLIT-TCP uses the standard TCP headers, flow and error controls, but it does not implement the standard congestion

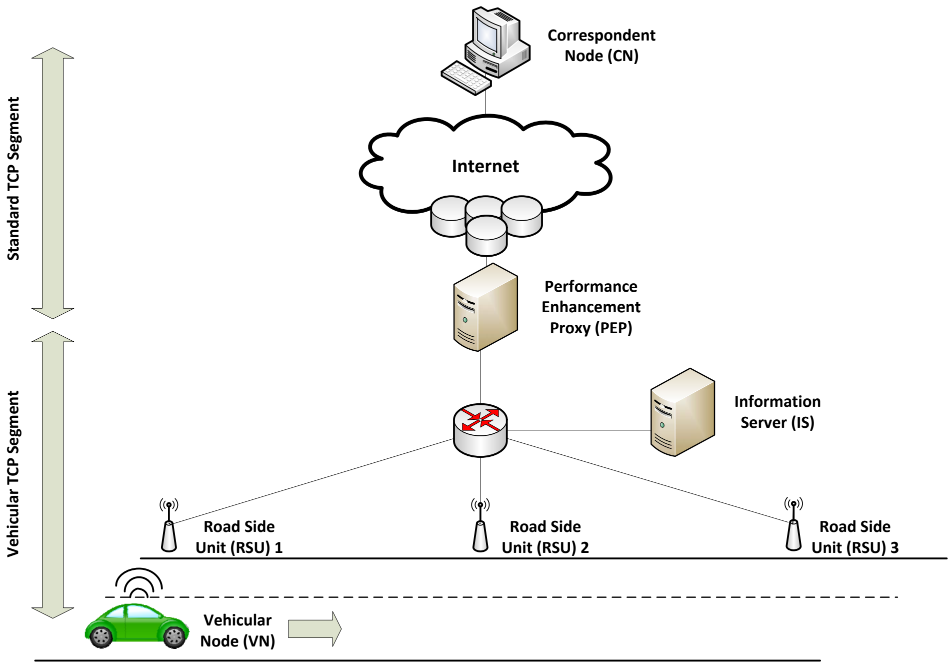

Fig. 5.1 TCP Handoff Architecture

control. The new congestion control algorithm is fed by the 802.21 MIH services installed in the wireless network to adapt the TCP communications to the situation of the candidate RSU when a vehicular user is going to perform a handoff.

5.2.1 Cross-layer design

Figure 5.1 shows the VSPLIT architecture. The CN is the remote node connected through a TCP connection to the VN. The standard TCP segment is from the CN to the PEP. From the PEP to the VN is considered the vehicular TCP Segment. VNs, RSUs and the IS implement the 802.21 MIH Function (MIHF). The PEP manages the different TCP connections, but it does not implement MIHF. The PEP is aware of the different handoffs by means of TCP Options (we define them in Section 5.2.4). We use an Information Server (IS) to provide Media Independent Information Service (MIIS) functionalities. This IS server has a holistic view of the network topology, and it knows the coordinates and coverage of all the RSUs in the network. In particular, we use the IS to provide the channel number of the candidate RSU to the VN when it is performing the handoff, avoiding large scanning times to find the next RSU.

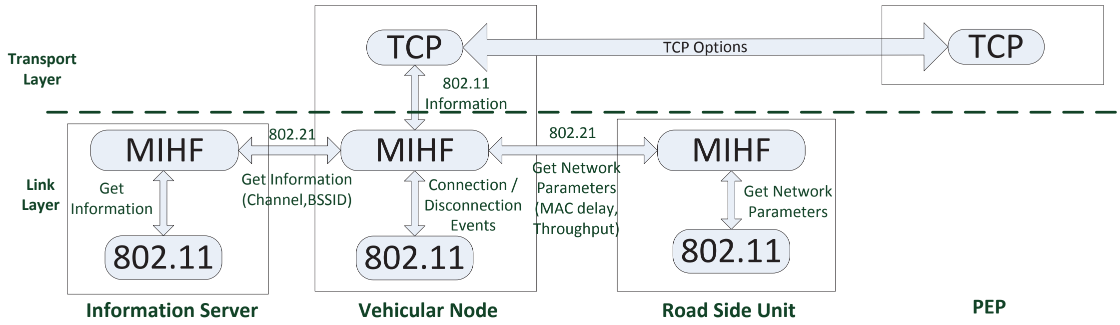

Fig. 5.2 Cross-layer information exchange

The cross-layer events will allow us to synchronize the link-layer connections/disconnections with the TCP flow (see Figure 5.2). The congestion window of active senders is set to a value that depends on the expected bandwidth for the next access network and the new RTT (we better explain this in Section 5.2.3). This is done just after the handoff when the VN is reconnected, without having to wait any TCP probing phase. These parameters are obtained from the RSU by the VN using 802.21. This modified congestion control can avoid packet loss, timeouts and spurious retransmissions, improving network efficiency and providing fairness among users.

The different speeds of vehicular nodes cause unfairness in TCP communications. Users that go faster generally achieve less throughput because they suffer a higher number of handoffs in the same amount of time than slow users. Also, the standard congestion control is going to need a certain time to reach the correct working point after the handoff, and this time is higher when there are other (slow) users that are transmitting at a high rate. Our congestion control is able to set the proper congestion window of all the TCP flows every time there is a new connection or disconnection in the RSU. This permits to reduce unfairness between fast and slow nodes as VSPLIT-TCP does not implement probing phases (slow start or congestion avoidance).

5.2.2 Design goals

The main goals of our VSPLIT architecture are:

- *Internet hosts must be able to use standard TCP*. The VSPLIT architecture achieves this goal by using one PEP (splitting). Since one of the TCP end users is the vehicular node, which is equipped with an On-Board Unit (OBU) with a particular protocol stack and with the 802.21 protocol, it is very feasible that the vehicular user can run a modified TCP protocol (in our case, the VSPLIT-TCP protocol). CNs do not require to be modified.

- *Full vehicular network exploitation.* This is an efficiency goal. TCP exhibits a bad performance in a vehicular network where a lot of handoffs are suffered. Our goal is to maximize the throughput of the competing TCP flows in front of losses, blackouts and bandwidth and delays variations.

- *Fair resource allocation among competing TCP flows.* TCP handoffs performed by vehicular users going at different speeds can produce unfairness behaviors. Those users who stay more time connected to the same RSU, in existing TCP approaches, get more throughput, since they suffer less handoffs and moreover there is no time for those users who go at high speeds to get the cwnd at the correct working point. Our handoff procedure can reduce the handoff disruption time and adapt the cwnd rapidly to the network situation, providing a better fairness behavior between TCP competing flows.

We propose an architecture based on TCP splitting, which consists in dividing the TCP connection into a vehicular network portion and a Internet portion. Using splitting, the connection is divided in two TCP segments: standard TCP Segment and vehicular TCP segment. Our architecture uses 802.11 Access Points (APs) as RSUs using each one different non-overlapped frequency channels. The 802.11 standard requires a MN to scan all the possible channels to discover available APs during a handoff. In a given 802.11 cell, several channels are expected to be empty. This makes the MN to waste a lot of time scanning empty channels, more than 90% of the overall handoff latency. The 802.11 scanning takes more than 300 ms to scan all the channels in typical 802.11 WLANs [80]. For this reason, avoiding scanning time is expected to improve the handoff latency and therefore minimize TCP communications disruptions. In this sense, our handoff architecture may be very useful as the VN can know in advance the Basic Services Set Identification (BSSID) and the current channel number of the candidate RSU.

5.2.3 VSPLIT-TCP Congestion Window Adaption

To maintain a link fully exploited, the cwnd of TCP senders must be set to the "Bandwidth-Delay Product" (BDP) [104]. When there is no competing traffic, the TCP flow should be able to obtain all the bandwidth at the bottleneck. In our scenario we assume that the bottleneck is the wireless domain. In the IEEE 802.11 MAC layer protocol, the sender has to contend to send only one data packet (and get an acknowledgment back) before contending for the channel again. This is clearly very different from the behavior of wired networks, where multiple packets can be sent without waiting for them to reach the other end of the link and being acknowledged. Therefore, the BDP of a TCP connection in 802.11 is smaller than in wired networks. However, when the MAC contention increases the calculated RTT by the TCP sender also increases and therefore the cwnd permits sending more packets. This behavior can overload the network because this RTT increase does not imply an increase in the capacity of the network.

To solve the aforementioned TCP cwnd overshooting problem, in [111] the authors split the total RTT into two parts: 1) congestion RTT and 2) contention RTT. The contention RTT is determined by the 802.11 MAC service time. This MAC service time is defined by the delay added to access the medium following the rules specified in the DCF algorithm. On the other hand, the remaining part of the RTT is the congestion RTT, which is determined by the transfer delay of all the links through the path excluding the contention RTT. The duration of continuous segment flow in the pipe is determined only by the congestion RTT. Therefore, BDP is determined by the congestion RTT and not by the contention RTT. If the original congestion-control mechanism [104] is directly applied over 802.11, the TCP cwnd is adjusted according to the total RTT (directly measured at the TCP agent). However, the congestion RTT, which is related to the network pipe volume, is smaller than the total RTT. As a result, the cwnd is likely to exceed the level of the BDP. Furthermore, the higher the ratio of the contention RTT to the congestion RTT is, the greater the overshooting problem becomes.

In [111], the authors propose a mechanism which provides an accurate method of estimating the contention status and adapts the cwnd to limit the window size from overshooting in multihop ad hoc networks. They deal with routing failure issues in static multihop networks. Multihop ad hoc network issues are different than TCP issues in V2I communications. Therefore our congestion control is different than the proposed by them. The idea is monitoring the throughput of a RSU to calculate the available bandwidth BW for each TCP flow. The following cwnd calculation will be used by the TCP agents in the PEP and the VN. A VN can estimate the available bandwidth by dividing the overall throughput in a RSU among the number of VN with active TCP connections in the cell. We use the MAC service time twice (go and return) as a $RTT_{contention}$ value. Notice that we consider that each RSU works in saturation mode, that is to say, there is always traffic to send. We will use Equations 5.1 and 5.2 to calculate the BDP:

$$BDP = (Tr_{RSU}/n) \times (RTT - RTT_{contention}) \tag{5.1}$$

$$BDP = (Tr_{RSU}/n) \times (RTT - (2 \times MAC\_service\_time)) \quad (5.2)$$

where Tr_{RSU} is the measured throughput in a RSU; n is the number of TCP users connected to the RSU; RTT is the round trip time calculated by the TCP agent; and the *MAC_service_time* is the MAC access contention delay calculated in the 802.11 network. The contention RTT is two times the *MAC_service_time*, because during the RTT the DCF algorithm acts two times (one time for the data packet, and another one for the acknowledgment), and subtracting it to the calculated RTT we obtain the congestion RTT. This *BDP* will be used to set the cwnd every time a handoff occurs. The entire handoff procedure is detailed in Section 5.3. Every time a vehicular node connects to a new RSU, it requests Tr_{RSU} and *MAC_service_time*, and therefore the PEP maintains them updated every handoff. Since in a vehicular scenario the number of nodes is high and, consequently, the number of handoffs in the network, the values used to calculate the cwnd will be updated frequently enough.

The strategy that our VSPLIT-TCP congestion control algorithm follows is to avoid congestion losses by construction, so all losses are due to transmission errors. For each handoff the cwnd for each user is recalculated (using Equation 5.1) to prevent congestion epochs. If the congestion control algorithm properly manages the system load, a packet loss does not have to involve a reduction of the packet injection rate, because losses are known for sure to be due to transmission errors. The VSPLIT-TCP sets the cwnd to the value equivalent to the BDP calculated by Equation 5.1. In case of errors, VSPLIT-TCP uses the mechanisms of standard TCP to retransmit lost packets.

5.2.4 VSPLIT-TCP Options

We defined three additional TCP options (see Figure 5.3) to inform the PEP about the handoffs of the VNs:

- TCP PreHandoff Option: VNs use it to communicate to the PEP that the node is going to disconnect from the current RSU. The TCP PreHandoff Option includes in the header the TCP Option Kind parameter and the length of the TCP Option.

- TCP PostHandoff Option: VNs use this option to communicate to the PEP that a node is connected to a new RSU, transmitting the network conditions of this new RSU, so the PEP is able to adapt all the TCP senders to the new characteristics of the communication. The TCP PostHandoff Option includes in the header the TCP Option Kind parameter, the length of the TCP Option, the MAC of the RSU that the VN is just connected and the network conditions used in Equation 5.1 to calculate the cwnd, that is, the RSU throughput (Tr_{RSU}) and the *MAC_service_time*.

- TCP Adaption Option: the PEP uses this option to communicate changes in the network parameters to VNs, for instance, when a VN node is connected or disconnected to a RSU. So, these VNs adapt their communication to the new situation. The TCP Adaption Option includes in the header the TCP Option Kind parameter, the length of the TCP Option and the new BDP calculated by the PEP for a specific RSU.

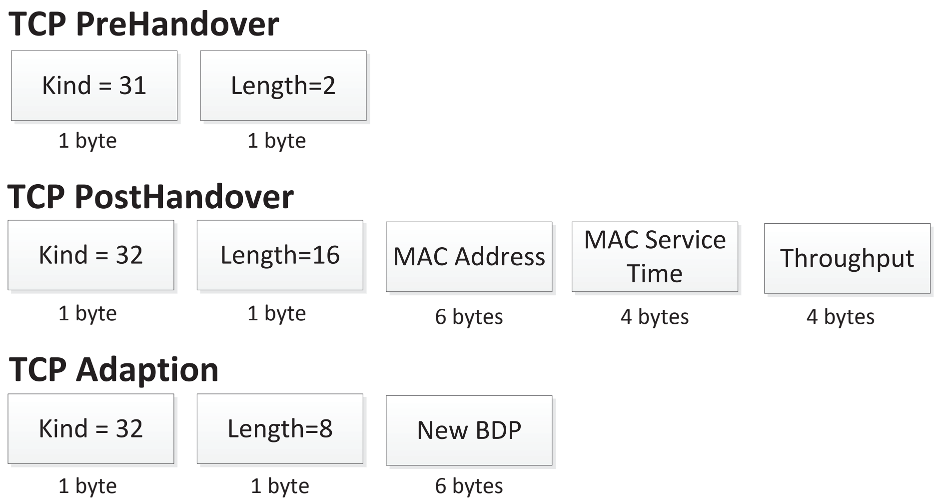

Fig. 5.3 VSPLIT-TCP Options

5.2.5 Live 802.11 Measurements

In the VSPLIT architecture, we use 802.21 services to inform the vehicular node about some live measurements of the candidate RSU. These live measurements are the 802.11 MAC Service Time and the 802.11 available throughput on each RSU. On the other hand, we also monitor the number of VNs connected to each RSU in order to calculate the BDP of each connection. The number of VNs connected to each RSU is monitored by the PEP, since in the 802.11p protocol there are no association or disassociation events when *dot11OCBEnabled* parameter is enabled (see Section 2.3), so it is difficult to know when a VN is connected to a RSU. Therefore, the PEP is responsible to know the number of VNs connected to a RSU, monitoring the TCP handoff messages used in the VSPLIT Handoff Procedure detailed in Section 5.3.

We consider a “saturation” condition, that is to say there is always traffic to send. This assumption may be easily acceptable in case the density of nodes connected to a RSU is relatively high. Assuming this, the traffic model generated by a single node does not affect to the entire RSU's bandwidth. In addition, if the number of vehicles is relatively high, we can also consider that the throughput for N users will be quite similar for $N+1$. Even that is not true, since the 802.11 performance decreases when the number of users connected to an access point is increased due to MAC contention, if the number of vehicles is relatively high this consideration is fairly good. Finally, notice that for each handoff the RSU TCP throughput is recalculated, thus the system always tends to achieve the proper working point.

802.11 MAC Service Time Calculation

To calculate the new RTT value after a wireless handoff, each RSU must collect the MAC service time. Using the difference between MAC service times after a handoff, a VN is able to calculate the new RTT of the new link and adapt the TCP algorithm to the new value rapidly. This prevents spurious timeouts or inefficient timers. In more detail, to obtain the current MAC service time in a 802.11 cell (MSR_{RSU}), we use a classical low-pass filter with $\alpha = 0.9$ (like the one that Van Jacobson used to calculate a smooth RTT value [104]). The equations are the following:

$$\begin{aligned} MSR_{RSU} &\leftarrow (1-\alpha)N + \alpha T \\ T &\leftarrow MSR_{RSU} \end{aligned} \tag{5.3}$$

Where N is the current measure of the MAC service time and T contains an accumulated value that is used to get a smooth value of MSR_{RSU}. In a vehicular scenario, the expected number of nodes is supposed to be high and, consequently, the number of handoffs. This will make the PEP to maintain the MSR_{RSU} updated frequently enough, since each time a node (vehicle) connects to a new RSU, the MSR_{RSU} value has to be recomputed to be provided to the vehicle.

802.11 Available Throughput

In VSPLIT, each RSU measures the overall TCP throughput that can be offered to its connected vehicles. This throughput is considered the available bandwidth at the bottleneck of the network. We assume that each RSU works in saturation mode. This total throughput will be shared between the amounts of users connected to the RSU (the bottleneck of the network), so new users can connect to the RSU using the total available bandwidth. In more detail, to obtain the current TCP throughput in an 802.11 cell (Tr_{RSU}), we use a classical low-pass filter with $\alpha = 0.9$. The equations are the following:

$$\begin{aligned} Tr_{RSU} &\leftarrow (1-\alpha)M + \alpha S \\ S &\leftarrow Tr_{RSU} \end{aligned} \tag{5.4}$$

Where M is the current measure of the TCP throughput and S contains an accumulated value that is used to get a smooth value of Tr_{RSU}. The Tr_{RSU} is also updated frequently enough, since each time a node (vehicle) connects to a new RSU, the Tr_{RSU} value has to be recomputed to be provided to the vehicle.

5.3 VSPLIT Handoff Procedure

In this section we detail the handoff procedure for TCP flows. We utilize a subset of existing IEEE 802.21 MIH services to enhance the handoff process in VSPLIT-TCP. The MIH service primitives used for the handoff decision making are:

- *MIH Get Information (MGI)*: This message is used by an MIHF to retrieve a set of Information Elements provided by the information service.
- *MIH Link Get Parameters (MLGP)*: This command is issued by upper layer entities to discover and monitor the status of the currently connected and potentially available links.
- *Link Handoff Imminent (LHI) / MIH Link Handoff Imminent (MLHI)*: This primitive is issued to report the imminent occurrence of an intra-technology link handoff. The *LHI* message is issued by the link layer and the *MLHI* message is issued by the MIHF.
- *Link Handoff Complete (LHC) / MIH Link Handoff Complete (MLHC)*: This primitive is issued to report the completion of an intra-technology link handoff. The *LHC* message is issued by the link layer and the *MLHC* message is issued by the MIHF.

Figure 5.4 shows the handoff procedure used for CN-to-VN TCP communications. We assume we work in a scenario where there are several RSUs deployed using non-overlapped frequency channels. The handoff procedure uses the VSPLIT-TCP protocol and it calculates the cwnd after a handoff using Equation 5.1. The TCP receiver placed at the VN acts as MIH user.

The procedure for CN-to-VN handoffs is detailed as follows:

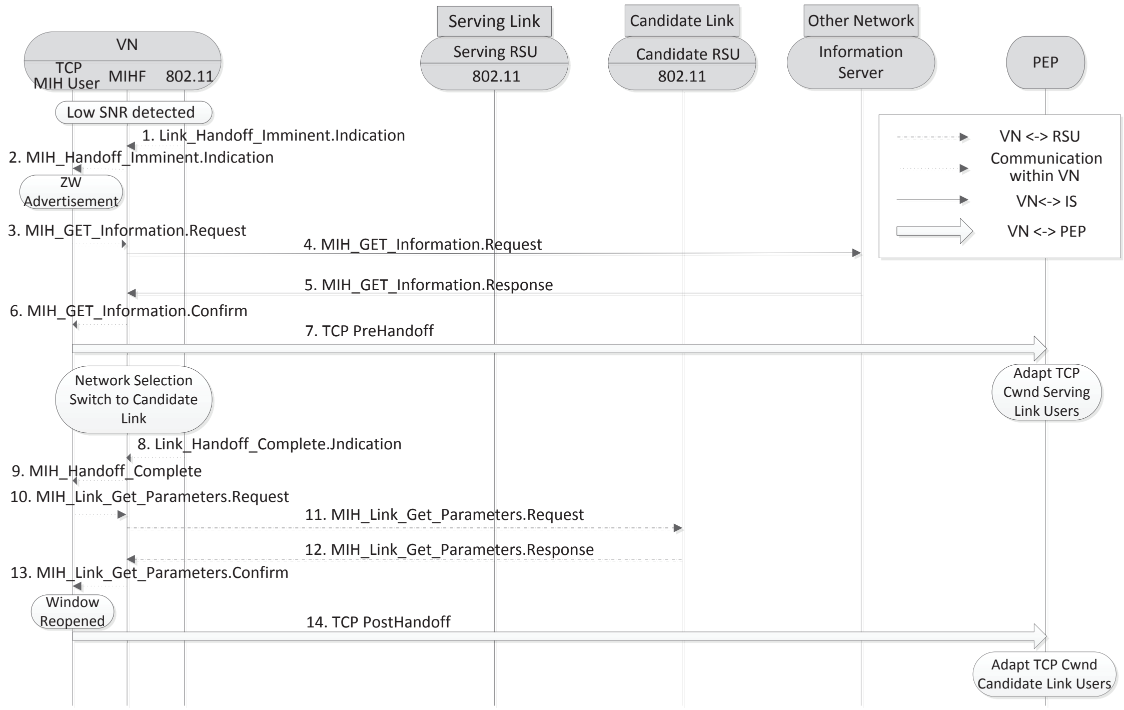

Fig. 5.4 VSPLIT-TCP CN-to-VN handoff

1. The VN is connected to the Serving RSU and it is receiving data segments from the CN. It detects that the signal-to-noise ratio (SNR) of the Serving RSU is below a certain threshold. Then the 802.11 layer sends a *LHI* indication message to the MIHF. This threshold depends on the speed of the node. The LHI must be at least an RTT before the *MLHC* indication. Therefore, if the node is slow, the threshold will be closer to the SNR value considered to switch to another RSU, than if the node goes faster.
2. The MIHF relays the handoff imminent information to the MIH User using the *MLHI* indication message. The TCP receiver is notified of the *MLHI* indication message and it sends a Zero Window (ZW) advertisement. When the TCP sender receives this ZW advertisement, the transmission is stopped.
3. At this point the VN MIH User sends a *MGI* request.
4. The MIHF relays the *MGI* request message to the IS. Using the requested information the VN can know the channel of the next RSU to connect (Candidate RSU).
5. The IS responds with the *MGI* response to the VN with the information requested.
6. The MIHF sends the *MGI* confirm to the VN MIH User.

7. The TCP receiver sends to the PEP a TCP Acknowledgment with the TCP PreHandoff Option included. Then the PEP is aware a node has left the Serving RSU and adapts the cwnd of the TCP flows of the nodes that are still connected to the Serving RSU, using Equation 5.1, to share immediately the bandwidth that has been released by the VN in the handoff.

8. The VN MIHF receives a *LHC* indication message after the communication is switched from the Serving RSU to the Candidate RSU.

9. The MIHF relays the *MLHC* indication message to the MIH User.

10. The VN MIH User sends a *MLGP* request to know the network conditions of this new RSU.

11. The VN MIHF relays the *MLGP* request to the Candidate RSU.

12. The VN MIHF receives the parameters from the Candidate RSU with a *MLGP* response (the RSU throughput Tr_{RSU} and the *MAC_service_time*).

13. The VN MIHF relays *MLGP* response to the VN MIH User.

14. The VN sends a TCP Acknowledgment to the PEP in the next segment with the TCP PostHandoff Option included. In this TCP PostHandoff Option the VN includes the parameters of the new RSU, so the PEP can calculate the cwnd using Equation 5.1. The PEP adapts the cwnd of the other VNs connected to the new RSU. The TCP Agents also adapt their retransmission counters to the new RTT calculated using the difference in the MAC service time in the last handoff. This avoids premature expiration of the RTO when the RTT varies abruptly.

Figure 5.5 shows the handoff procedure used for VN-to-CN TCP communications. Here, the VNs act as TCP senders and the PEP as a TCP receiver. Therefore, it is the same VN who will adapt the cwnd with the new RSU parameters, but the VN will have to send the parameters to the PEP, and the PEP will broadcast them to the other VNs to be able to adapt their connections. The TCP sender placed at the VN acts as MIH user. The procedure for VN-to-CN handoffs is detailed as follow:

1. The VN is connected to the serving network via the current RSU and it is receiving data segments from the CN. It detects that the SNR of the current RSU is below a certain threshold. Then the 802.11 layer sends a *LHI* indication message to the MIHF.

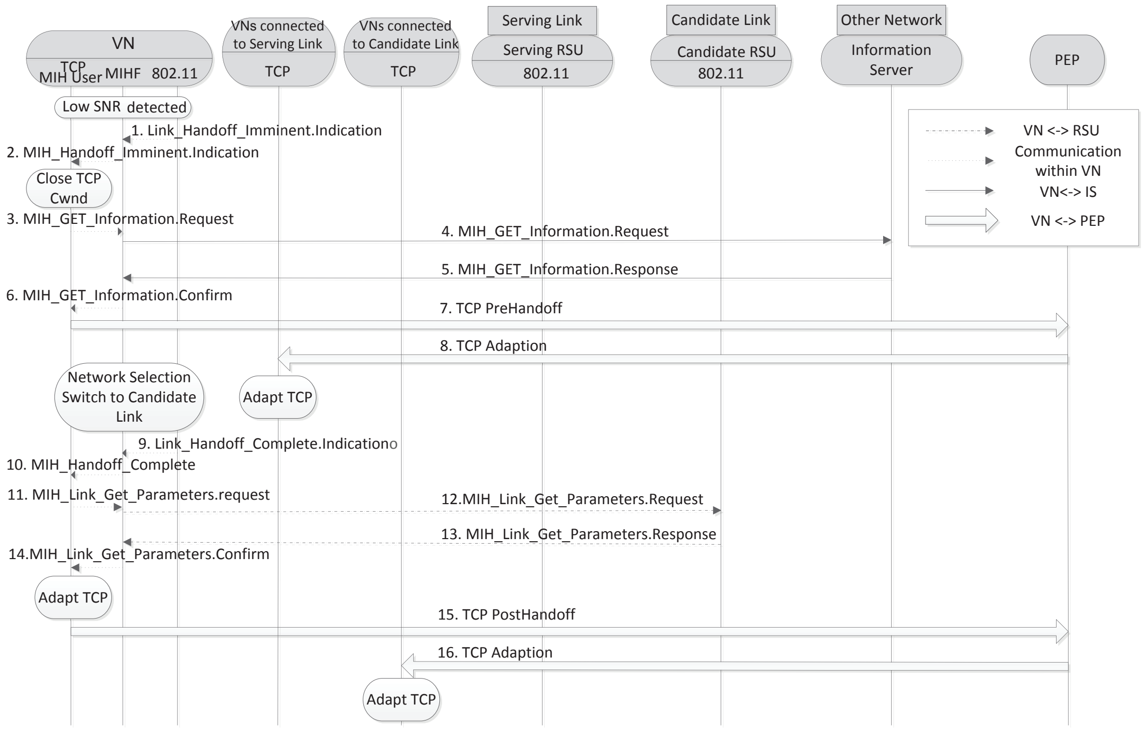

Fig. 5.5 VSPLIT VN-to-CN handoff

2. The MIHF relays the handoff imminent information to the MIH User using the *MLHI* indication message. When the TCP sender receives the *MLHI* indication, the VN will stop its transmission. The cwnd is set to 0.
3. At this point the VN MIH User sends a *MGI* request.
4. The MIHF relays the *MGI* request message to the IS.
5. The IS responds with the *MGI* response to the VN with the information requested.
6. The MIHF sends the *MGI* confirm to the VN MIH User.
7. The VN sends a TCP message with the TCP PreHandoff Option included to the PEP.
8. When the PEP receives the TCP PreHandoff Option, it includes the TCP Adaption Option to the TCP Acknowledgment messages for the other VNs connected to the Serving RSU to adapt their cwnd to share immediately the bandwidth that has been released by the VN in the handoff.
9. The VN MIHF receives a *LHC* indication message after the communication is switched from the Current RSU to the Candidate RSU.

10. The MIHF relays the *MLHC* indication message to the MIH User.

11. The VN MIH User sends a *MLGP* request to know the network state of this new RSU.

12. The VN MIHF relays the *MLGP* request to the Candidate RSU.

13. The VN MIHF receives the parameters from the Candidate RSU with a *MLGP* response.

14. The VN MIHF relays the *MLGP* message to the VN MIH User.

15. The VN TCP sender adapts its cwnd to the new parameters following Equation 5.1 and its retransmission timers and restarts the communication. It sends a TCP segment to the PEP with the TCP PostHandoff Option included with the parameters of the new RSU.

16. The PEP attaches the TCP Adaption option to the TCP Acknowledgements for the VNs connected to the Candidate RSU to adapt its cwnd using Equation 5.1 with the new BDP calculated.

Link (or network) layer handoff procedures are implied in the handoff architecture and the figures does not represent these messages. That is to say that the procedure represented is for link-layer handoffs, but with a very slight modification it can be applied to network-layer handoffs. To be used using network-layer handoffs (e.g, Mobile IP [88]), the only modification that must be done is that the TCP agent placed at the VN must wait for the Binding Acknowledgment from the CN to send the PostHandoff Option, indicating when the TCP client is able to transmit again.

5.4 Performance evaluation

The vehicular scenario designed to test the performance of our architecture is a highway with three lanes per direction, with the characteristics of the scenario depicted in Figure 5.1. This is an infrastructure scenario where a set of RSUs are deployed over a highway in an overlapped manner. Therefore there are no coverage blackouts in the road. All the RSUs are connected to a central router and this is also connected to the PEP. The RSUs belong to the same subnet, so every handoff in the scenario is a layer 2 handoff. There is a communication between a fixed node in the infrastructure (CN) and the vehicular nodes. We analyzed the performance during the communications for CN-to-VN flows and VN-to-CN flows. We analyzed the aggregated throughput of the sum of all the vehicular nodes TCP flows and the fairness obtained between them using the architecture described in Section 5.2. We compared the VSPLIT-TCP approach with the Freeze-TCP approach and with the TCP NewReno [59]. We used TCP NewReno as a reference for a TCP implementation without cross-layer modifications for handoffs, and Freeze-TCP as a reference for a horizontal TCP handoff approach. We did not compared the performance evaluation of vertical handoff approaches detailed in Section 5.1 because they do not apply to our reference scenario.

To perform the evaluation we implemented the proposed architecture and the TCP modifications in the ns-3 simulator [10]. We used the Simulation of Urban MObility (SUMO) [36] as a road traffic mobility simulator to generate the mobility traces used by ns-3 to model the mobility of the vehicular nodes. We implemented a 802.11 MIH_LINK_SAP that interacts with the 802.21 MIH Function of the nodes. We also implemented the Freeze-TCP [58] approach to be able to compare our solution with another approach that tries to solve TCP mobility issues. In our Freeze-TCP implementation we used the *MIH Link Handoff Imminent* and the *MIH Link Handoff Complete* 802.21 messages as connection and disconnection signals, because in the original article there is no specification about how to handle with link layer signals.

Parameter Name	Value
Wired links	Bandwidth: 100Mb Propagation delay: 2ms
Propagation model	Two-Ray Ground
Interface queue	Droptail
Distance between RSUs	200m
Packet size	1500 bytes
Highway scenario	2000 meters
Number of RSUs	10
Simulation Duration	200 s

Table 5.2 Simulation parameters

In Table 5.2 the simulation parameters are detailed. The simulation time is 200 seconds and the length of highway is about 2000 meters. We used the MAC and PHY parameters detailed in [22]. The parameters used are depicted in Table 5.3. We used a basic rate of 3Mbps and a data rate of 27Mbps. The propagation model used is the Two-Ray Ground. We used four different speeds for the vehicles. These speeds are 20, 25, 30 and 35 m/s.

Parameter Name	Value
Slot time	13 μs
SIFS	32 μs
DIFS	58 μs
RTS/CTS	Enabled
Rx Threshold	-82 dBm
CS Threshold	-86 dBm
Tx Power Level	35 dBm
Data rate	27 Mbps
Basic rate	3 Mbps
Intended range	250 m

Table 5.3 MAC and PHY 802.11 parameters

5.4.1 Throughput

In Figures 5.6 and 5.7 we can observe the aggregated throughput of all the vehicular nodes moving along the road. The aggregated throughput is obtained with the addition of the throughput of all the TCP flows of the vehicular nodes in the simulation. We evaluate the scenario for different number of contending users in the simulation, from 25 to 200 users. We can observe the improvement in the aggregated throughput is up to the 10%. We can also observe that the behavior of our proposal when the number of vehicles increase is better than the other solutions. As it can be observed, the other protocols (Freeze-TCP and TCP-NewReno) are more affected by congestion when increasing the number of vehicles in the simulation, in contrast with VSPLIT that avoids it by using the cross-layer congestion window mechanism detailed in Section 5.2.3.

In Figure 5.8 and 5.9 we can observe the average throughput per vehicle, using different speeds. We used 100 nodes in this simulation. We can observe that our proposal performs better for most of the cases, specially at high speeds. This behavior is due to vehicles at high speeds suffer more handoffs, so Freeze-TCP and TCP-NewReno need more time to arrive to an optimal working point than our proposal. However, there are some cases (only for the VN-to-CN flow and at slow speed) in which Freeze-TCP and TCP-NewReno have a better performance than our architecture, mainly because classical congestion window algorithms perform better when there are few handoffs.

Throughput is important, but other aspects should also be evaluated to determine the suitability of a TCP flavor. In particular, fairness is crucial. As we will demonstrate in the rest of this section, our architecture shares in a better way the available bandwidth between users. The other solutions do not recalculate the congestion window when a new connection (or disconnection) occurs, so a user already connected with a high congestion window (and

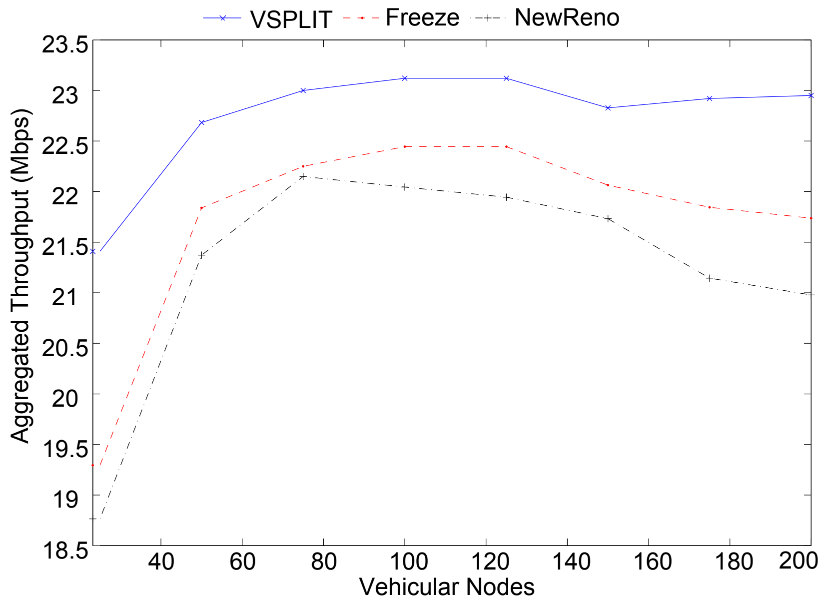

Fig. 5.6 Aggregated throughput CN-to-VN

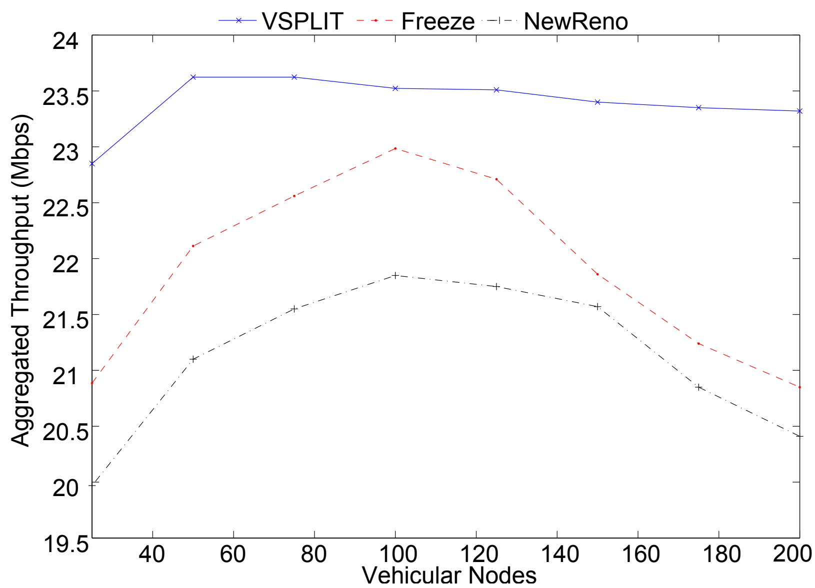

Fig. 5.7 Aggregated throughput VN-to-CN

therefore a great bandwidth consumption) can create problems to newly connected users, showing unfairness behaviors.

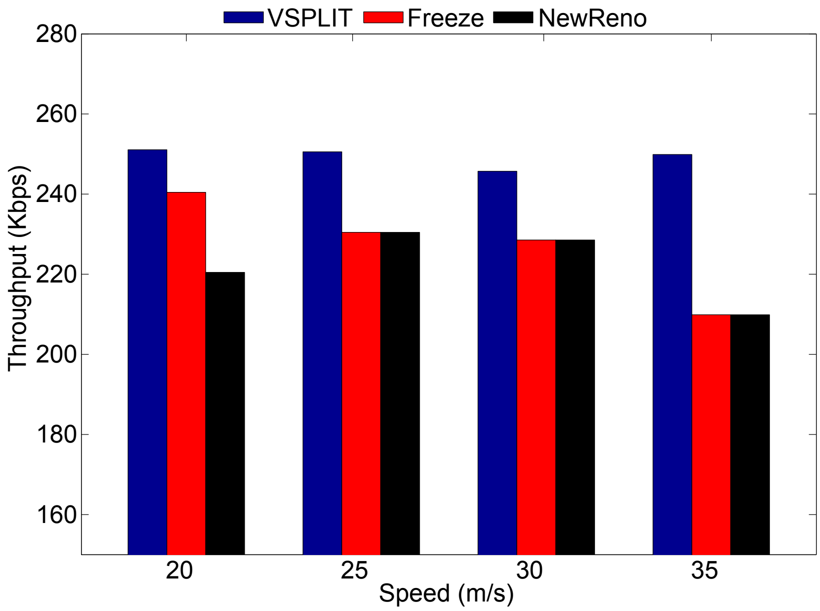

Fig. 5.8 Average throughput per speeds CN-to-VN

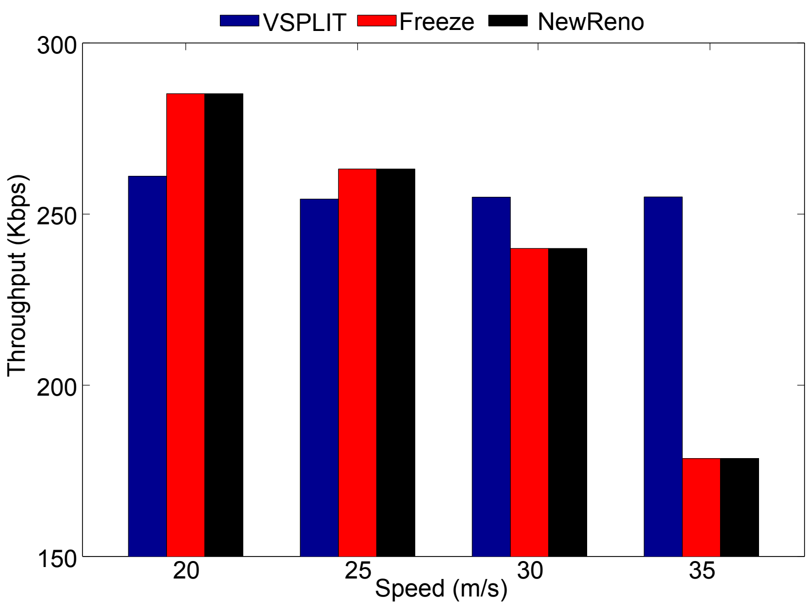

Fig. 5.9 Average throughput per speeds VN-to-CN

5.4.2 Throughput Fairness

The *TCP-fairness* level represents a generic term that describes the ability of a TCP variant to coexist with the same TCP variant, fairly sharing the available bandwidth in the same bottleneck link. Throughput fairness is an important criterion for evaluating the proper working of any TCP. We used the Jain's fairness index [63] to evaluate the throughout fairness among users. This fairness index is always between 0 and 1. A lower value implies poorer fairness. If all throughputs are the same, the fairness index is 1.

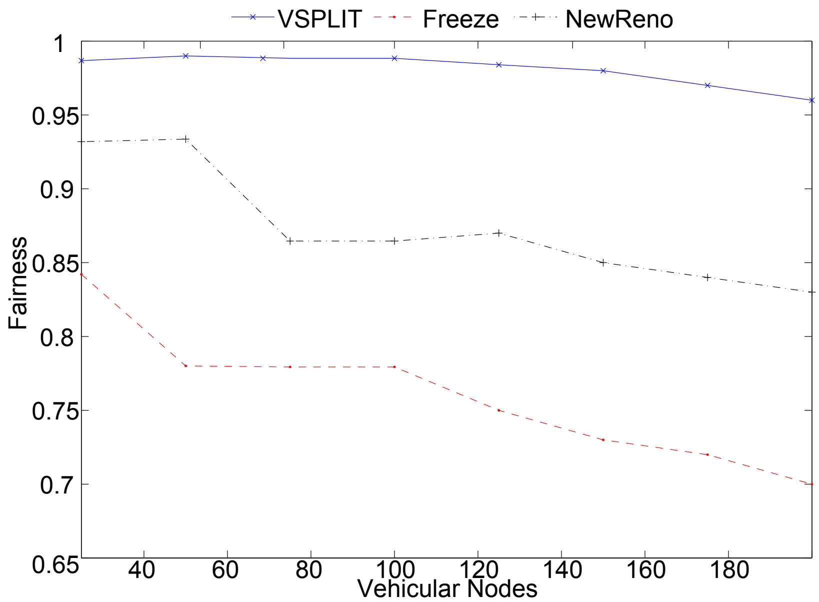

Fig. 5.10 Fairness CN-to-VN

Figure 5.10 and Figure 5.11 show the fairness between all the TCP flows in the evaluated scenario. Fairness is represented as a function of the number of vehicular users in the simulation. From the figures it can be observed that the fairness index for VSPLIT remains almost constant independently from the number of nodes, meanwhile Freeze-TCP and TCP NewReno are more severely affected and they show a decreasing behavior as the number of vehicles in the simulation increases. While Freeze-TCP and TCP NewReno get a fairness index between 0.85 and 0.7 for CN-to-VN communications and an index between 0.82 and 0.8 for VN-to-CN communications, VSPLIT-TCP never falls to a value less than 0.95.

5.5 Conclusions

Numerous TCP handoff approaches have been proposed in the literature to solve TCP issues during handoffs. In this chapter, we have summarized and analyzed different approaches

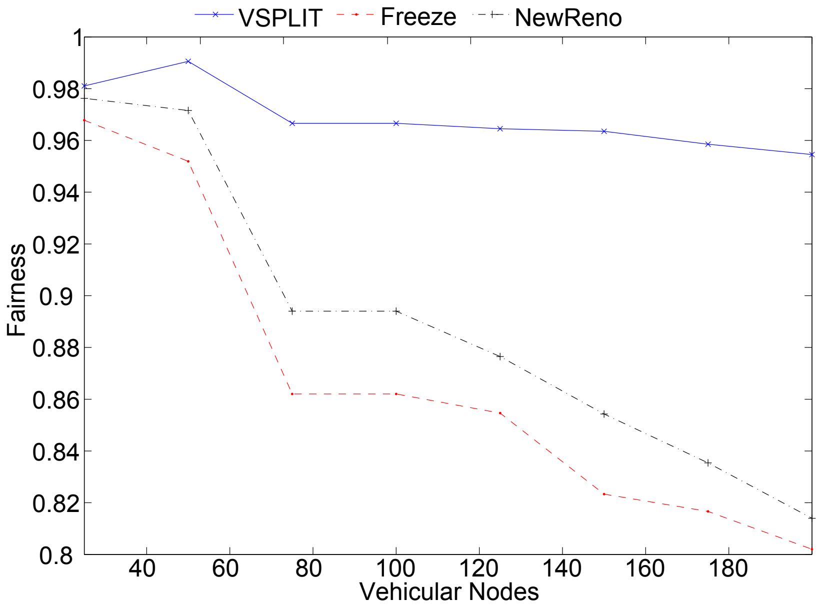

Fig. 5.11 Fairness VN-to-CN

oriented to solve TCP issues during horizontal or vertical handoffs. However, none of these approaches is oriented to vehicular handoffs in 802.11 networks. TCP congestion-control algorithms in 802.11 networks cause an overshooting window problem and result in poor throughput, due to MAC contention in 802.11 channels cause an increment of the RTT that does not imply an increase in the capacity of the network. Also, handoffs disruptions can cause unfairness between vehicular nodes going at different speeds.

In this chapter we have presented a new TCP-aware handoff architecture that works using the IEEE 802.21 MIH standard. This architecture provides feedback information about the network situation of the RSUs to the TCP transport layer, just at the moment of the handoff is presented. This feedback information provides to the transport layer the ability to adapt the congestion window to the new network parameters minimizing the convergence speed of the congestion control algorithm to the characteristics of the new link. The proposed architecture has been designed for 802.11 networks and can deal with layer 2 and layer 3 handoffs. The proposal has been implemented in the ns-3 simulator and a set of simulations are presented. These simulations demonstrate the good performance of the proposed architecture in a vehicular scenario with handoffs between 802.11 RSUs.

Chapter 6

Conclusions and Further Work

ITS is a hot research topic and is attracting a great interest in the automotive and telecommunications industry. Vehicular communications has attracted a lot of research driven by public and private organizations, but mainly oriented to enhance safety in the transportation network. Nowadays there is no infrastructure installed and available to use by vehicles, so this will be an issue to deploy new vehicular services. Non-safety applications can play a vital role for the successful deployment of this vehicular networks infrastructure. Infotainment applications can be an impulse not only for users, but also for network operators because they can provide new and interesting business opportunities that will promote the necessary investment in infrastructure.

We have shown in this thesis that handoffs management is an important problem to deal with, and specially relevant for multimedia applications and services with real-time requirements. We have presented three contributions to tackle with handoffs management when using V2I infotainment services. First, we proposed a vehicular simulator based on emulation to provide a testing tool for real-time infotainment services. Second, using the previous tool, we evaluated video services over vehicular networks using Mobile IP and FMIP protocols. And finally, we proposed a new TCP architecture that uses IEEE 802.21 cross-layer information to help during the handoff procedure.

The remainder of this chapter summarizes the main results from our research and identifies some future research lines.

6.1 Conclusions

In this thesis, we have designed and implemented a vehicular emulation platform named Vehicular EmulationS Platform for Real Applications (VESPA). VESPA provides a fast and cheap environment that allows testing real infotainment applications using vehicular traffic mobility. VESPA integrates a network simulator with emulation features (ns-2), a road traffic mobility simulator (SUMO) and UML (User Mode Linux) virtual machines, all these configurable by using a GUI tool.

VESPA, in contrast with existing simulation platforms, is mainly focused on providing a testing framework for V2I infotainment applications. VESPA allows to recreate vehicular scenarios in an easy way, and to test real Linux-based applications thanks to the use of UML virtual machines. So, it is possible to test applications using the same software developed for desktop computers without lasting time in modeling these applications for network simulators, avoiding the limitations caused by simplified application behaviors. Also, using the emulation feature, VESPA is able to test applications with real-time requirements, such as multimedia applications. In addition, VESPA is the only vehicular testing tool that supports Mobile IP and FMIP protocols. We have demonstrated that the accuracy of VESPA is comparable to the ns-2 network simulator, and it is more scalable in terms of CPU utilization or ratio of late packets than other emulation tools.

For these and many other reasons, VESPA could be an interesting and useful tool for researchers and applications developers, who can test their applications and protocols in realistic vehicular scenarios in a straightforward manner. In this sense, VESPA can be an excellent tool to test OBUs software without being worried about all the parameters of lower networking layers. For example, VESPA can be used to check video services deployment in the vehicular environment, or to compare the performance of different video codification techniques, or to compare the performance of the different TCP flavor implementations available in the Linux kernel.

As a key proof of the usefulness of VESPA, we have presented a set of simulations of live video streaming over vehicular networks in Chapter 4. Video applications and video contents are expected to be a growing infotainment service, such as real-time traffic information broadcasting or various on-road video entertainments (live sports, news, etc.). In this set of experiments, we have analyzed how handoffs (both Mobile IP and FMIP) limit the overall quality of a video streamed during a trip over a highway. We have analyzed a real-time video solution using UDP+RTP protocols, emphasizing in the QoE perceived by the user (using the PSNR of the decoded video), and the disruption times in handoffs during the video playback.

After these tests, we have observed that the packet loss rate grows as the video bitrate and the vehicle speed increases. We can state that packet losses limit the deployment of video streaming services in vehicular networks at high speeds, even in case of using fast handoffs techniques (FMIP), which were supposed to minimize handoffs blackouts. At low speeds (20 and 30 m/s), predictive FMIP handoffs are predominant and the video streaming service can be played normally. However, at high speeds (40 and 50 m/s) reactive handoffs are predominant and the number of disruptions and the disruption times are so high that avoid a proper reproduction of the video service. In fact, at these speeds the behavior of Mobile IP and FMIP is very similar. This can be an issue for potential customers that are willing to pay for video services in car trips, and it can be more important in countries where the law does not limit the overall speed at certain highways.

We can foresee some solutions to reduce losses during handoffs. For instance, one possibility may be to use advanced video coding techniques, such as application-layer FEC techniques or SVC extension of H.264/MPEG4-AVC. Another possibility to reduce or to eliminate losses during handoffs may be to use a reliable TCP connection. TCP may seem unsuitable for some real-time services, but popular videoconference or video streaming services like Skype, YouTube or Netflix use TCP as transport protocol. TCP is known to suffer from a bad performance when there is intermittent connectivity, for instance caused by handoffs, so the third contribution of this thesis is focused in this problem.

In this third contribution we pretend to deal with handoff management in V2I communications. We have presented VSPLIT, a new TCP-aware handoff architecture that uses the IEEE 802.21 MIH standard to provide feedback information to the transport layer at the moment of the handoff. VSPLIT is a TCP-splitting architecture where a modified TCP (VSPLIT-TCP) protocol is used between the vehicular users and the PEP, meanwhile the Internet nodes use a standard TCP flavor. IEEE 802.21 MIH services are used to provide cross-layer information to VSPLIT-TCP, so it is possible to adapt the congestion window of TCP flows to the characteristics of the new link, and minimizing the convergence speed of the congestion control algorithm. The VSPLIT architecture has been designed for 802.11 horizontal handoffs, and it can deal with layer 2 and layer 3 handoffs. We have implemented VSPLIT in the ns-3 simulator, demonstrating its good performance in a vehicular scenario with handoffs between 802.11 RSUs, specially at high speeds. VSPLIT gets up to 43% more throughput than other approaches at high speeds, maintaining fairness.

6.2 Further Work

In this section we explore possible improvements and open research directions based on ideas and results provided in this dissertation.

6.2.1 VESPA new release goals

We are still working on improving VESPA. One feature that would be interesting to provide to VESPA is the possibility of using different types of virtual machines. TAP virtual interfaces not only permit the connection of UML virtual machines, but also different ones such as KVM [5] or VirtualBox [16]. The idea is letting the user to choose not only the application software, but also the operating system and the virtualization tool to be used. There is also the possibility of integrating the Virtual Network User Mode Linux (VNUML) [55] tool with VESPA. VNUML provides an interesting feature in order to configure in an easy way the network devices of each UML machine using an XML file, and also to control the application commands executed on each virtualized node.

In Section 3.4 we studied the accuracy and scalability of VESPA, and we realized that there is a tradeoff between scalability and accuracy. This is an important issue when testing large-scale networks, such as vehicular networks. To improve the scalability of VESPA, we plan to include some implementation enhancements. We pretend to include distributed virtualization capabilities to VESPA. This means that VESPA would be able to distribute the virtualization of nodes through a group of hosts. As a possible candidate for a virtualization architecture, it is possible to use EDIV [56], which can be considered a distributed version of the virtualization tool VNUML [55], as it allows the transparent management of distributed scenarios.

6.2.2 Transport-layer handoffs

In this thesis we have demonstrated that network and link-layer handoffs in V2I communications cause disruptions that affect the performance of infotainment services, for both UDP (see Chapter 4) and TCP (see Chapter 5). For this reason, we think that it would be a good idea to use a transport-layer mobility approach, instead of a network or link-layer mobility approach. When using a transport-layer mobility approach, it is possible to take advantage of multihoming capabilities, reducing or eliminating losses during handoffs and disruption times. Multihoming is defined as the capacity of a node to be connected to different networks, using multiple interfaces. This would allow the vehicular node to accumulate more than one transport-layer connection to the destination, and to migrate from one to another data transport flow when necessary, avoiding sharp disconnections.

Using transport-layer mobility, no new hardware or software component is necessary in the infrastructure. We consider that the use of transport-layer mobility would achieve a lower handoff latency and packet loss rate, if compared to network-layer mobility. It would also be more efficient in terms of routing, since it avoids inefficient routing paths caused by the triangle routing issue [87]. Transport-layer mobility can also help to avoid conflicts with network security solutions such as ingress filtering and firewalls. Many routers or firewalls discard packets coming from the internal network if the packets do not contain a source IP address configured for one of the internal networks, and mobile nodes use their home address as the source IP address of the packets they transmit, when using Mobile IP. Transport-layer mobility also increases the survivability and scalability, in contrast with network-layer mobility. In Mobile IP, the home agent must intercept all packets between a mobile node and a correspondent node, and when the home network is vulnerable to failure, it can generate problems. It is difficult to replicate the home agent at various locations in order to achieve survivability. Handling mobility at the transport layer is a promising approach to achieve seamless handoff in the context of heterogeneous wireless access networks [38]. Transport-layer mobility supports a change of the IP address on the underlying network layer, while keeping the end-to-end connection alive. It also permits the use of different access networks concurrently, with the benefits it offers in terms of bandwidth.

During the last two years, the MPTCP (Multipath Transport Control Protocol) [34] working group of the IETF has been developing multipath extensions to TCP [52]. These extensions will enable hosts to use several paths, possibly through multiple interfaces, to carry the packets that belong to a single connection. User experience can be improved using MPTCP when vehicular nodes use multiple interfaces either simultaneously or alternatively while moving along the road. However, there are many technical challenges to be considered before being able to provide seamless interoperability when using different access networks. Some of these challenges are packet reordering, retransmissions, congestion window managing, etc. In particular, we plan to work in enhancing the performance of video infotainment services in vehicular networks when using MPTCP as transport protocol, following two different research lines. The first one will deal with scheduling for the MPTCP protocol, and the second one pretend to include some modifications in the MPTCP protocol to take into account multimedia QoS semantics to improve video performance using Deep Packet Inspection (DPI) [2]. In more detail:

- **Class-based MPTCP scheduler for delay-sensitive traffic.** Several solutions have been implemented in order to take advantage of multihoming and multipath capabilities of mobile nodes [53, 82]. However, one major problem of multipath transfer in multihomed networks is the utilization of different paths with different bandwidths and delays. Jitter degradation and the receiver buffer blocking problem [62] occur when multiple paths have disparate performance characteristics. Any multipath transport protocol design must establish how to send data over the available subflows. Therefore, the design of the scheduler is critical for the efficient operation of MPTCP, because it performs the distribution of the individual packets of an application flow over several available subflows. A scheduler can take into account several variables, for example the buffer size, the capacity, or the delay of each subflow. MPTCP does not specify any scheduler in the standard. So, we pretend to work on a class-based scheduler for a delay-sensitive traffic in a multipath scenario. The objective of this scheduler is to find a flow distribution among the different paths that minimizes the average latency, and providing priority scheduling between different QoS services.

- **Autonomous Multipath Communication Architecture for infotainment multimedia applications**. We will work on a new multipath architecture designed as a part of the MPTCP layer in order to guarantee the QoS to infotainment mobile applications. The idea is to take full advantage of the intrinsic multimedia QoS semantics based in DPI in order to self-manage the available resources and to provide a more compliant e2e transport service in multihomed vehicular networks. This architecture pretend to integrate Autonomic Computing [68] control including monitoring, analysis, planning and execution phases, and it will integrate a common knowledge base composed of media semantics and the required policies and strategies to guarantee self-management properties.

Nomenclature

Acronyms / Abbreviations

ACID Application Class Identifier

ACM Application Context Mark

AU Application Unit

BDP Bandwidth-Delay Product

C2C Car-to-Car

$C2C-CC$ CAR 2 CAR Communication Consortium

CN Correspondent Node

cwnd congestion window

DSRC Dedicated Short-Range Communications

ETSI European Telecommunications Standards Institute

FMIP Fast Handoff for Mobile IP

IEEE Institute of Electrical and Electronics Engineers

IETF International Engineering Task Force

IP Internet Protocol

ITS Intelligent Transportation Systems

MIPv6 Mobile IPv6

MN Mobile Node

NEMO NEtwork MObility

OBU On-Board Unit

RAT Radio Access Technology

RSU Road Side Unit

RTO Retransmission TimeOut

RTT Round-Trip Time

SME Station Management Entity

$STRAW$ Street Random Waypoint

$SUMO$ Simulation of Urban MObility

$SWANS$ Scalable Wireless Ad Hoc Network Simulator

TCP Transport Control Protocol

$TIGER$ Topologically Integrated Geographic Encoding and Referencing system

$TraNS$ Traffic and Network Simulation Environment

UDP User Datagram Protocol

$V2I$ Vehicle-to-Infrastructure

$V2V$ Vehicle-to-Vehicle

$VeiNS$ Vehicles in Network Simulation

$VESPA$ Vehicular EmulationS Platform for real Applications

VN Vehicular Node

$WAVE$ Wireless Access in Vehicular Environment

References

[1] ISO TC204 WG16 CALM. http://www.isotc204wg16.org/.

[2] ITU-T Y.2770 (11/2012) Requirements for deep packet inspection in next generation networks.

[3] Source code for ns-extension on HMIPv6 with Fast-Handover. http://mobqos.ee.unsw.edu.au/~robert/nsinstall.php.

[4] IEEE 2006 IEEE 1609 Trial-Use Standard for Wireless Access in Vehicular Environments (WAVE).

[5] KVM - Kernel Based Virtual Machine. http://www.linux-kvm.org/page/Main_Page.

[6] MPEG-2 Encoder / Decoder, Version 1.2, July 19, 1996. http://www.mpeg.org/MSSG/.

[7] MPlayer - The Movie Player. http://www.mplayerhq.hu.

[8] NO Ad-Hoc Routing Agent (NOAH). http://icapeople.epfl.ch/widmer/uwb/ns-2/noah/.

[9] Network Simulator, version 2. http://www.isi.edu/nsnam/ns.

[10] ns-3 Official Website. http://www.nsnam.org.

[11] OMNet++ Network Simulation Framework. http://www.omnetpp.org.

[12] OpenStreetMap. http://www.openstreetmap.org.

[13] VTun - Virtual Tunnels over TCP/IP networks. http://vtun.sourceforge.net.

[14] Tiger Map Service. http://www.census.gov/geo/www/tiger/.

[15] Temporally-Ordered Routing Algorithm. http://wiki.uni.lu/secan-lab/Temporally-Ordered+Routing+Algorithm.html.

[16] Oracle VM VirtualBox. https://www.virtualbox.org/.

[17] Veins - Vehicles in Network Simulation. http://veins.car2x.org/.

[18] VideoLan - VLC media player - Open Source Multimedia Framework and Player. http://www.videolan.org/vlc/.

[19] Iso/iec standard for information technology - telecommunications and information exchange between systems - local and metropolitan area networks - specific requirements part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications (includes ieee std 802.11, 1999 edition; ieee std 802.11a.-1999; ieee std 802.11b.-1999; ieee std 802.11b.-1999/cor 1-2001; and ieee std 802.11d.-2001). *ISO/IEC 8802-11 IEEE Std 802.11 Second edition 2005-08-01 ISO/IEC 8802 11:2005(E) IEEE Std 802.11i-2003 Edition*, pages 1–721, 2005. doi: 10.1109/IEEESTD.2005.339589.

[20] IEEE Standard for Information Technology - Telecommunications and Information Exchange between systems - Local and Metropolitan Area Networks - Specific Requirements. *IEEE Std 802.11e-2005 (Amendment to IEEE Std 802.11)*, pages 1 –189, November 2005. doi: 10.1109/IEEESTD.2005.97890.

[21] Ieee standard for local and metropolitan area networks- part 21: Media independent handover. *IEEE Std 802.21-2008*, pages c1 –301, 21 2009. doi: 10.1109/IEEESTD.2009.4769367.

[22] Ieee standard for information technology– local and metropolitan area networks– specific requirements– part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 6: Wireless access in vehicular environments. *IEEE Std 802.11p-2010*, pages 1 –51, 15 2010. doi: 10.1109/IEEESTD.2010.5514475.

[23] Standard Specification for Telecommunications and Information Exchange Between Roadside and Vehicle Systems — 5 GHz Band Dedicated Short Range Communications (DSRC) Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *ASTM E2213 - 03(2010)*, 2010. doi: 10.1520/E2213-03R10.

[24] ETSI EN 302 665, Intelligent Transport Systems (ITS); Communications Architecture, 2010.

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

[26] VANET-based optimization of infotainment and traffic efficiency vehicular services, 2012.

[27] ITS-Safety 2010. Its-safety 2010, 2009. URL http://wiki.fot-net.eu/index.php?title=ITS-Safety_2010.

[28] S. Adibi. *Quality of Service Architectures for Wireless Networks: Performance Metrics and Management*. IGI Global, 2010. ISBN 9781615206810.

[29] S. Ahmad, R. Hamzaoui, and M. Al-Akaidi. Adaptive unicast video streaming with rateless codes and feedback. *Circuits and Systems for Video Technology, IEEE Transactions on*, 20(2):275 –285, feb. 2010. ISSN 1051-8215. doi: 10.1109/TCSVT.2009.2031545.

[30] S. Allal and S. Boudjit. Geocast routing protocols for vanets: Survey and guidelines. In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on*, pages 323–328, July 2012. doi: 10.1109/IMIS.2012.133.

[31] J. Arkko, C. Vogt, and W. Haddad. Enhanced Route Optimization for Mobile IPv6. RFC 4866 (Proposed Standard), May 2007. URL http://www.ietf.org/rfc/rfc4866.txt.

[32] R. Baldessari, C.J. Bernardos, and M. Calderon. Geosac - scalable address autoconfiguration for vanet using geographic networking concepts. In *Personal, Indoor and Mobile Radio Communications, 2008. PIMRC 2008. IEEE 19th International Symposium on*, pages 1–7, Sept 2008. doi: 10.1109/PIMRC.2008.4699949.

[33] R. Barr. Java in Simulation Time / Scalable Wireless Ad hoc Network. http://jist.ece.cornell.edu.

[34] Sébastien Barré, Christoph Paasch, and Olivier Bonaventure. Multipath tcp: From theory to practice. In *IFIP Networking, Valencia*, May 2011.

[35] S.A. Baset and H.G. Schulzrinne. An analysis of the skype peer-to-peer internet telephony protocol. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–11, April 2006. doi: 10.1109/INFOCOM.2006.312.

[36] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz. Sumo - simulation of urban mobility: An overview. In *SIMUL 2011, The Third International Conference on Advances in System Simulation*, pages 63–68, Barcelona, Spain, October 2011.

[37] P. Bhagwat and CE Perkins. Highly dynamic destination-sequenced distance vector routing (DSDV) for mobile computers. In *Proceedings of ACM SIGCOMM*, volume 94, pages 234–244, 1994.

[38] Ł. Budzisz, R. Ferrús, A. Brunstrom, K.-J. Grinnemo, R. Fracchia, G. Galante, and F. Casadevall. Towards transport-layer mobility: Evolution of {SCTP} multihoming. *Computer Communications*, 31(5):980 – 998, 2008. ISSN 0140-3664. doi: http://dx.doi.org/10.1016/j.comcom.2007.12.014. URL http://www.sciencedirect.com/science/article/pii/S014036640700549X. Mobility Management and Wireless Access.

[39] C2C-CC. CAR 2 CAR Communication Consortium. http://www.car-to-car.org/.

[40] R. Caceres and L. Iftode. Improving the performance of reliable transport protocols in mobile computing environments. *Selected Areas in Communications, IEEE Journal on*, 13(5):850 –857, jun 1995. ISSN 0733-8716. doi: 10.1109/49.391749.

[41] Qi Chen, Felix Schmidt-Eisenlohr, Daniel Jiang, Marc Torrent-Moreno, Luca Delgrossi, and Hannes Hartenstein. Overhaul of ieee 802.11 modeling and simulation in ns-2. In *Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems*, MSWiM '07, pages 159–168, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-851-0. doi: 10.1145/1298126.1298155. URL http://doi.acm.org/10.1145/1298126.1298155.

[42] David Choffnes and Fabián E. Bustamante. Straw - an integrated mobility and traffic model for vanets. In *Proceedings of The 10th International Command and Control Research and Technology Symposium (CCRTS)*, June 2005.

[43] CICAS. Cooperative intersection collision avoidance systems (cicas), 2009. URL http://www.its.dot.gov/cicas/.

[44] CVIS. Cooperative vehicule-intrastructure system, 2010. URL http://www.cvisproject.org/.

[45] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert. Network Mobility (NEMO) Basic Support Protocol. RFC 3963 (Proposed Standard), January 2005. URL http://www.ietf.org/rfc/rfc3963.txt.

[46] J. Dike et al. The user-mode linux kernel homepage. *URL: http://user-mode-linux.sourceforge.net*.

[47] J.C. Duenas, F. Cuadrado, B. Garcia, H.A. Parada G, and J.L. Ruiz. System virtualization tools for software development. *Internet Computing, IEEE*, 13(5):52 –59, sept.-oct. 2009. ISSN 1089-7801. doi: 10.1109/MIC.2009.115.

[48] S. Eichler. Performance evaluation of the IEEE 802.11 p WAVE communication standard. In *2007 IEEE 66th Vehicular Technology Conference, 2007. VTC-2007 Fall*, pages 2199–2203, 2007.

[49] R. Fernandes, P.M. d'Orey, and M. Ferreira. Divert for realistic simulation of heterogeneous vehicular networks. In *Mobile Adhoc and Sensor Systems (MASS), 2010 IEEE 7th International Conference on*, pages 721–726, Nov 2010. doi: 10.1109/MASS.2010.5663806.

[50] R. Fernandes, F. Vieira, and M. Ferreira. Vns: An integrated framework for vehicular networks simulation. In *Vehicular Networking Conference (VNC), 2012 IEEE*, pages 195–202, Nov 2012. doi: 10.1109/VNC.2012.6407431.

[51] R. Finlayson. Live555 streaming media, 2009. http://www.live555.com/liveMedia/.

[52] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure. TCP Extensions for Multipath Operation with Multiple Addresses. RFC 6824 (Experimental), January 2013. URL http://www.ietf.org/rfc/rfc6824.txt.

[53] Shaojian Fu, Liran Ma, M. Atiquzzaman, and Yong-Jin Lee. Architecture and performance of sigma: a seamless mobility architecture for data networks. In *Communications, 2005. ICC 2005. 2005 IEEE International Conference on*, volume 5, pages 3249–3253 Vol. 5, May 2005. doi: 10.1109/ICC.2005.1495024.

[54] Zhenghua Fu, P. Zerfos, Haiyun Luo, Songwu Lu, Lixia Zhang, and M. Gerla. The impact of multihop wireless channel on tcp throughput and loss. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 3, pages 1744–1753 vol.3, March 2003. doi: 10.1109/INFCOM.2003.1209197.

[55] F. Galán, D. Fernándeza, J. Ruiz, O. Walid, and T. de Miguel. A virtualization tool in computer network laboratories. In *Proceedings of 5th International Conference on Information Technology Based Higher Education and Training (ITHET'04)*, volume 17, pages 27–34, Istanbul, May 2004.

[56] F. Galan, D. Fernandez, J. Vergara, and F. Monserrat. Demo of ediv: Building and managing distributed virtualization scenarios in federated testbed infrastructures. In *5th International Conference on Testbeds(TridentCom 2009)*, pages 1 –3, april 2009. doi: 10.1109/TRIDENTCOM.2009.4976229.

[57] G. Giambene, S. Marchi, and S. Kota. Cross-layer schemes for tcp performance improvement in hetnets for high-speed trains. In *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, pages 1 –6, dec. 2011. doi: 10.1109/GLOCOM.2011.6134138.

[58] T. Goff, J. Moronski, D.S. Phatak, and V. Gupta. Freeze-tcp: A true end-to-end tcp enhancement mechanism for mobile environments. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1537–1545. IEEE, 2000.

[59] T. Henderson, S. Floyd, A. Gurtov, and Y. Nishida. The NewReno Modification to TCP's Fast Recovery Algorithm. RFC 6582 (Proposed Standard), April 2012. URL http://www.ietf.org/rfc/rfc6582.txt.

[60] ISO. International organization for standardization, 2014. URL http://www.iso.org/.

[61] E. Ivov, J. Montavont, and T. Noel. Thorough empirical analysis of the ietf fmipv6 protocol over ieee 802.11 networks. *Wireless Communications, IEEE*, 15(2):65 –72, april 2008. ISSN 1536-1284. doi: 10.1109/MWC.2008.4492979.

[62] J.R. Iyengar, P.D. Amer, and R. Stewart. Receive buffer blocking in concurrent multipath transfer. In *Global Telecommunications Conference, 2005. GLOBECOM '05. IEEE*, volume 1, pages 6 pp.–, 2005. doi: 10.1109/GLOCOM.2005.1577365.

[63] Rajendra K. Jain, Dah-Ming W. Chiu, and William R. Hawe. A Quantitative Measure Of Fairness And Discrimination For Resource Allocation In Shared Computer Systems. Technical report, Digital Equipment Corporation, September 1984.

[64] Jangkeun Jeong, Hyuntai Kim, Taeyoung Lee, and Jitae Shin. An analysis of hidden node problem in ieee 802.11 multihop networks. In *Networked Computing and Advanced Information Management (NCM), 2010 Sixth International Conference on*, pages 282–285, Aug 2010.

[65] Daniel Jiang, Vikas Taliwal, Andreas Meier, Wieland Holfelder, and Ralf Herrtwich. Design of 5.9 GHz DSRC-based vehicular safety communication. *IEEE Wireless Communications*, 13(5):36–43, 2006.

[66] Xiaoxiao Jiang, Xiang Cao, and D.H.C. Du. Multihop transmission and retransmission measurement of real-time video streaming over dsrc devices. In *A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2014 IEEE 15th International Symposium on*, pages 1–9, June 2014. doi: 10.1109/WoWMoM.2014.6918955.

[67] D. Johnson, Y. Hu, and D. Maltz. The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4. RFC 4728 (Experimental), February 2007. URL http://www.ietf.org/rfc/rfc4728.txt.

[68] J.O. Kephart and D.M. Chess. The vision of autonomic computing. *Computer*, 36 (1):41–50, Jan 2003. ISSN 0018-9162. doi: 10.1109/MC.2003.1160055.

[69] M.Q. Khan and S.H. Andresen. Zero scanning time for 802.11 networks by using media independent information server (miis). In *Advanced Information Networking and Applications (AINA), 2012 IEEE 26th International Conference on*, pages 467–473, March 2012. doi: 10.1109/AINA.2012.100.

[70] Jirka Klaue, Berthold Rathke, and Adam Wolisz. Evalvid - a framework for video transmission and quality evaluation. In Peter Kemper and William H. Sanders, editors, *Computer Performance Evaluation / TOOLS*, volume 2794 of *Lecture Notes in Computer Science*, pages 255–272. Springer, 2003. ISBN 3-540-40814-2. URL http://dblp.uni-trier.de/db/conf/cpe/cpe2003.html#KlaueRW03.

[71] R. Koodli. Mobile IPv6 Fast Handovers. RFC 5568 (Proposed Standard), July 2009. URL http://www.ietf.org/rfc/rfc5568.txt.

[72] R. Koodli and C. Perkins. Mobile IPv4 Fast Handovers. RFC 4988 (Experimental), October 2007. URL http://www.ietf.org/rfc/rfc4988.txt.

[73] V. Kumar, Lan Lin, D. Krajzewicz, F. Hrizi, O. Martinez, J. Gozalvez, and R. Bauza. itetris: Adaptation of its technologies for large scale integrated simulation. In *Vehicular Technology Conference (VTC 2010-Spring), 2010 IEEE 71st*, pages 1–5, 2010. doi: 10.1109/VETECS.2010.5494182.

[74] Fan Li and Yu Wang. Routing in vehicular ad hoc networks: A survey. *Vehicular Technology Magazine, IEEE*, 2(2):12–22, June 2007. ISSN 1556-6072. doi: 10.1109/MVT.2007.912927.

[75] Bin Liu, Philippe Martins, and Philippe Bertin. Cross-layer design of the inter-rat handover between umts and wimax. *EURASIP Journal on Wireless Communications and Networking*, 2010:1–13. URL http://www.hindawi.com/journals/wcn/2010/763614/.

[76] M. Luglio, C. Roseti, G. Savone, and F. Zampognaro. Cross-layer architecture for a satellite-wi-fi efficient handover. *Vehicular Technology, IEEE Transactions on*, 58 (6):2990 –3001, july 2009. ISSN 0018-9545. doi: 10.1109/TVT.2008.2011274.

[77] H. Hartenstein M. Torrent Moreno, A. Festag. System design for information dissemination in vanets. In *3rd International Workshop on Intelligent Transportation (WIT)*, pages 27–33, March 2006.

[78] Daniel Mahrenholz. Adjusting the ns-2 emulation mode to a live network. In *KiVS 2005: Fachtagung Kommunikation in Verteilten Systemen*, pages 205–217. Springer, 2005.

[79] Daniel Mahrenholz and Svilen Ivanov. Real-Time Network Emulation with ns-2. In *Proceedings of DS-RT*, pages 29–36, 2004.

[80] Arunesh Mishra, Minho Shin, and William Arbaugh. An empirical analysis of the ieee 802.11 mac layer handoff process. *SIGCOMM Comput. Commun. Rev.*, 33(2): 93–102, April 2003. ISSN 0146-4833. doi: 10.1145/956981.956990. URL http://doi.acm.org/10.1145/956981.956990.

[81] NOW. Network on wheels, 2007. URL https://dsn.tm.kit.edu/english/projects_now-project.php.

[82] Christoph Paasch, Gregory Detal, Fabien Duchene, Costin Raiciu, and Olivier Bonaventure. Exploring mobile/wifi handover with multipath tcp. In *Proceedings of the 2012 ACM SIGCOMM Workshop on Cellular Networks: Operations, Challenges, and Future Design*, CellNet '12, pages 31–36, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1475-6. doi: 10.1145/2342468.2342476. URL http://doi.acm.org/10.1145/2342468.2342476.

[83] California PATH. California path, 2012. URL http://www.path.berkeley.edu/about.

[84] C. Perkins. IP Mobility Support for IPv4, Revised. RFC 5944 (Proposed Standard), November 2010. URL http://www.ietf.org/rfc/rfc5944.txt.

[85] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561 (Experimental), July 2003. URL http://www.ietf.org/rfc/rfc3561.txt.

[86] C. Perkins, D. Johnson, and J. Arkko. Mobility Support in IPv6. RFC 6275 (Proposed Standard), July 2011. URL http://www.ietf.org/rfc/rfc6275.txt.

[87] C.E. Perkins. Mobile networking through mobile ip. *Internet Computing, IEEE*, 2 (1):58–69, Jan 1998. ISSN 1089-7801. doi: 10.1109/4236.656077.

[88] C.E. Perkins. Mobile IP. *International Journal of Communication Systems*, 11(1): 3–20, 1998.

[89] M. Piórkowski, M. Raya, A.L. Lugo, P. Papadimitratos, M. Grossglauser, and J.P. Hubaux. TraNS: realistic joint traffic and network simulator for VANETs. *ACM SIGMOBILE Mobile Computing and Communications Review*, 12(1):31–33, 2008.

[90] COMeSafety Project. Communications for esafety (comesafety), 2013. URL http://www.comesafety.org/.

[91] ERTICO Project. Ertico - its europe, 2010. URL http://www.ertico.com/ertico-its-europe/.

[92] Smartway Project. National institute for land and infrastructure management, japan, 2009. URL http://www.nilim.go.jp/japanese/its/3paper/pdf/060131trb.pdf.

[93] Vijay T. Raisinghani and Tata Infotech. Eclair: An efficient cross layer architecture for wireless protocol stacks. In *In World Wireless Congress, SF*, page 2004, 2004.

[94] Ashwin Rao, Arnaud Legout, Yeon-sup Lim, Don Towsley, Chadi Barakat, and Walid Dabbous. Network characteristics of video streaming traffic. In *Proceedings of the Seventh COnference on Emerging Networking EXperiments and Technologies*, CoNEXT '11, pages 25:1–25:12, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-1041-3. doi: 10.1145/2079296.2079321. URL http://doi.acm.org/10.1145/2079296.2079321.

[95] Sergi Reñé, C. Gañán, Juan Caubet, J. Alins, J. Mata, and J. L. Muñoz. Analysis of video streaming performance in vehicular networks. The First International Conference on Advanced Communications and Computation, 10/2011 2011. ISBN 978-1-61208-161-8.

[96] Sergi Rene, Juanjo Alins, Jorge Mata-Diaz, Carlos Ganan, Jose L. Munoz, and Oscar Esparza. Vespa: Emulating infotainment applications in vehicular networks. *IEEE Pervasive Computing*, 13(3):58–66, 2014. ISSN 1536-1268. doi: http://doi.ieeecomputersociety.org/10.1109/MPRV.2014.60.

[97] Sergi Reñé, Oscar Esparza, Juanjo Alins, Jorge Mata-Díaz, and JoseL. Muñoz. Vsplit: A cross-layer architecture for v2i tcp services over 802.11. *Mobile Networks and Applications*, 18(6):831–843, 2013. ISSN 1383-469X. doi: 10.1007/s11036-013-0473-8. URL http://dx.doi.org/10.1007/s11036-013-0473-8.

[98] SafeTrip21. Safetrip21, 2009. URL http://wiki.fot-net.eu/index.php?title=SafeTrip21.

[99] T. Schierl, K. Ganger, C. Hellge, T. Wiegand, and T. Stockhammer. Svc-based multisource streaming for robust video transmission in mobile ad hoc networks. *Wireless Communications, IEEE*, 13(5):96 –103, october 2006. ISSN 1536-1284. doi: 10.1109/WC-M.2006.250365.

[100] SEVECOM. Secure vehicular communication, 2008. URL http://www.esafetysupport.info/download/related_projects/sevecom.pdf.

[101] A.K. Singh and S. Iyer. Atcp: Improving tcp performance over mobile wireless environments. In *Mobile and Wireless Communications Network, 2002. 4th International Workshop on*, pages 239 – 243, 2002. doi: 10.1109/MWCN.2002.1045729.

[102] J.P. Singh, N. Bambos, B. Srinivasan, and D. Clawin. Wireless LAN performance under varied stress conditions in vehicular traffic scenarios. In *2002 IEEE 56th Vehicular Technology Conference, 2002. Proceedings. VTC 2002-Fall*, volume 2, 2002.

[103] F. Soldo, C. Casetti, C. Chiasserini, and P.A. Chaparro. Video streaming distribution in vanets. *Parallel and Distributed Systems, IEEE Transactions on*, 22(7):1085–1091, July 2011. ISSN 1045-9219. doi: 10.1109/TPDS.2010.173.

[104] W.R. Stevens and G.R. Wright. *TCP/IP Illustrated: the protocols*, volume 1. Addison-Wesley, 1994.

[105] Guolin Sun and Bo Hu. Delay analysis of ieee 802.11 dcf with back-off suspension. In *Future Generation Communication and Networking (FGCN 2007)*, volume 1, pages 148–151, Dec 2007. doi: 10.1109/FGCN.2007.95.

[106] Vikas Taliwal, Daniel Jiang, Heiko Mangold, Chi Chen, and Raja Sengupta. Empirical determination of channel characteristics for dsrc vehicle-to-vehicle communication. In *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, VANET '04, pages 88–88, New York, NY, USA, 2004. ACM. ISBN 1-58113-922-5.

[107] S. Thomson, T. Narten, and T. Jinmei. IPv6 Stateless Address Autoconfiguration. RFC 4862 (Draft Standard), September 2007. URL http://www.ietf.org/rfc/rfc4862.txt.

[108] Vaishampayan, R. and Garcia-Luna-Aceves, J.J. volume , pages 304–313, oct. 2004. doi: {}.

[109] S.Y. Wang and C.L. Chou. Nctuns tool for wireless vehicular communication network researches. *Simulation Modelling Practice and Theory*, 17(7): 1211–1226, 2009. ISSN 1569-190X. doi: DOI:10.1016/j.simpat.2009.04.008. URL http://www.sciencedirect.com/science/article/B6X3C-4W6Y11C-1/2/8f6278d0c7948335c2151ad101a39fe7.

[110] Jiang Xie, I. Howitt, and I. Shibeika. Ieee 802.11-based mobile ip fast handoff latency analysis. In *Communications, 2007. ICC '07. IEEE International Conference on*, pages 6055–6060, 2007. doi: 10.1109/ICC.2007.1003.

[111] Xin Ming Zhang, Wen Bo Zhu, Na Na Li, and Dan Keun Sung. Tcp congestion window adaptation through contention detection in ad hoc networks. *Vehicular Technology, IEEE Transactions on*, 59(9):4578 –4588, nov. 2010. ISSN 0018-9545. doi: 10.1109/TVT.2010.2070522.

[112] Junlan Zhou, Zhengrong Ji, and Rajive Bagrodia. Twine: A hybrid emulation testbed for wireless networks and applications. In *In IEEE INFOCOM 2006*, pages 23–29. Society Press, 2006.

Appendix A

Support to Mobile IP and FMIP

As network layer mobility protocols, in Chapter 4, we use the IETF standards Mobile IP and FMIP. In this section we explain the handoff procedure for Mobile IPv4, Mobile IPv6 and FMIPv6.

A.1 Mobile IPv4 handoff

Mobile IPv4 [84] is a protocol that allows a mobile node (MN) to maintain its connectivity while changing its point of attachment to the Internet. The protocol uses two IP addresses: (1) one address called Home Address (HoA) that is permanently assigned to the MN from its Home Agent (HA), and it is used as identifier; and (2) another address called Care of Address (CoA), used for routing and assigned by the Foreign Agent (FA) in the range of addresses this FA manages.

The Mobile IPv4 handoff process has four main phases: movement detection, configuration, registration and information transfer (see Figure A.1). HA and FA advertise their presence via *Agent Advertisement* messages so that they become known by the MN. With that information, the MN can determine whether it is in its home network or in a foreign network. A MN can optionally solicitate an *Agent Advertisement* message sending an *Agent Solicitation* request to receive the advertisement. When an MN detects that it has moved to a foreign network, it obtains a CoA address. This CoA can be allocated by the FA itself (the one present in the subnetwork), or by any alternative mechanism such as DHCP. This address is used to identify the MN in the local network. Once the address is obtained, the MN updates its HA. The MN sends a *Registration Request* message to its FA and the FA relays the message to the HA. The HA replies with a *Registration Reply* message. At this time packets sent to the MN's HoA are intercepted and tunneled to the MN's CoA by the HA, received at the tunnel endpoint and finally delivered to the MN at its current location.

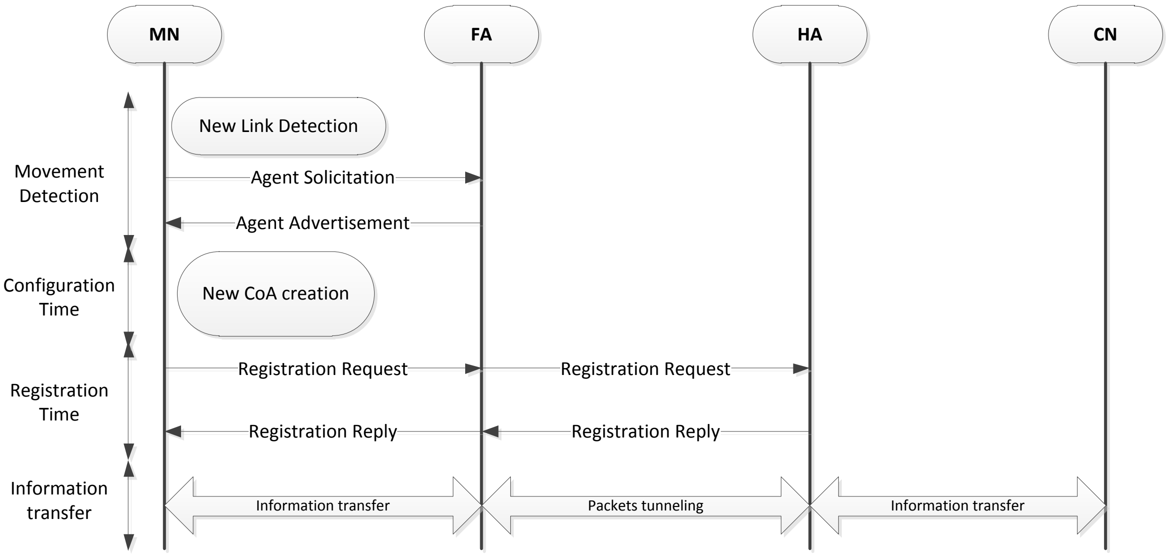

Fig. A.1 Mobile IPv4 Handoff

When the MN moves from one subnet to another, it registers its new CoA from the new location to its HA. After the registration, the HA has to receive packets addressed to the MN. If the FA is the tunnel endpoint, it desencapsulates the packets and forwards them to the MN. If the tunnel endpoint is the MN, it only has to desencapsule the packets.

Mobile IP requires that an MN sends a location update to its HA whenever it moves from one subnet to another. This location registration delay is long, which results in long handoff delay and high packet loss during the handoff process. During the handoff process in Mobile IP, packets are dropped until the new connection is established even though the MN could still communicate with its CN via the old AP.

A.2 Mobile IPv6 handoff

There are some changes from Mobile IPv4 [84] to Mobile IPv6 [86] mainly due to the differences between IPv4 and IPv6. The CoA can be automatically allocated on the visited network due to the new address autoconfiguration feature of IPv6 [107]. Because of the enormous address space of IPv6, an MN can acquire a co-located CoA on any foreign link quickly and easily. As a result, the FA function is abandoned in Mobile IPv6. The HA keeps a binding between a pair of IPv6 addresses used to map the HoA of an MN onto its current CoA. Route Optimization [31] is embedded in Mobile IPv6 so that packets can be transmitted from a correspondent node (CN) to an MN directly. As a result, an MN needs to periodically send update messages not only to the HA, but also to its CN.

The Mobile IPv6 handoff procedure includes movement detection, configuration, registration and information transfer, as shown in Figure A.2. An MN can detect the migration to a new subnet by analyzing a message periodically sent by the Access Router (AR) called *Router Advertisement* (RA).

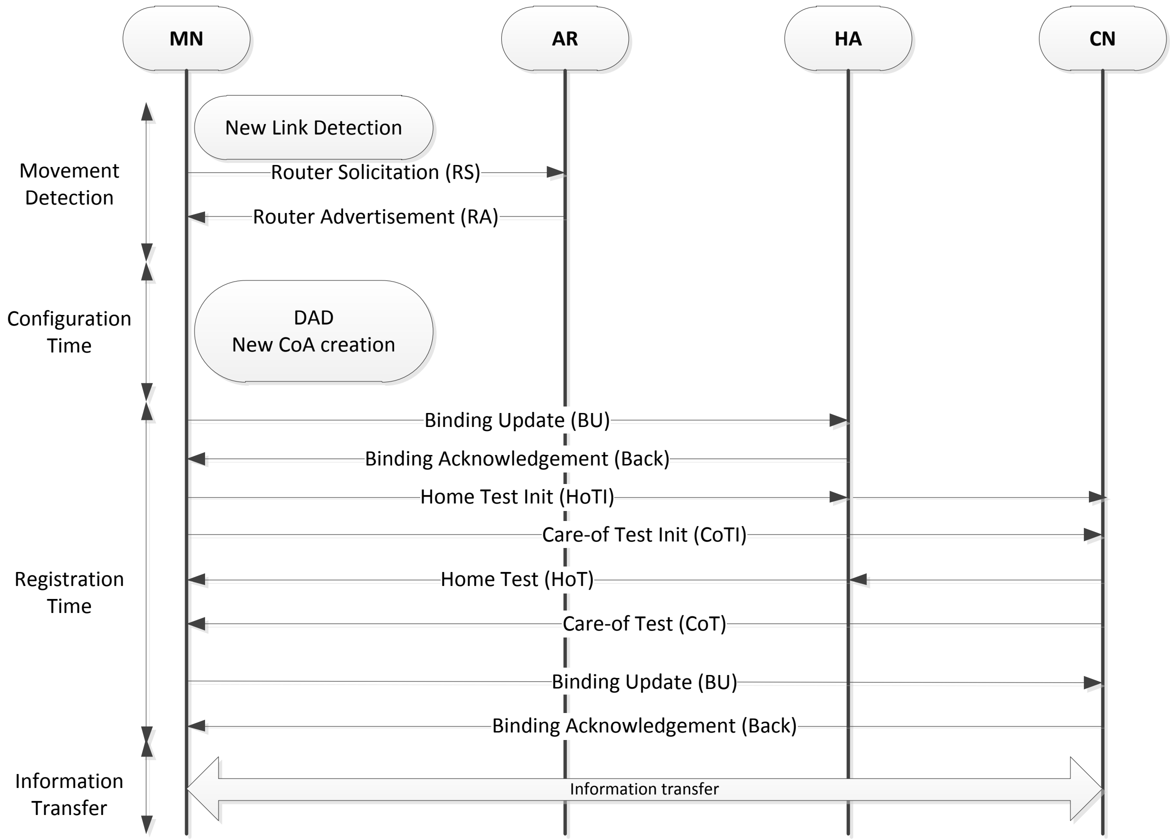

Fig. A.2 Mobile IPv6 Handoff

Using the information contained inside the RA, the MN is able to create a new CoA using the address autoconfiguration feature. After the address autoconfiguration, the MN must perform duplication address detection (DAD) to verify the uniqueness of the address on the new link. A *Binding Update* (BU) message is defined to provide mobility support, and it combines the functions of the *Registration Request* of Mobile IPv4 (see Figure A.1) and the message for *Route Optimization*. Therefore, the MN entering in a new subnet updates its location at the HA and at the CN by exchanging *Binding Acknowledgments* (Back) with both. The return routability process that provides route optimization consists of two checks, a HoA check and a CoA check to guarantee the legitimacy of the MN. This procedure is based on the exchange of four messages with the CN prior to send the BU message. The MN sends to the CN two messages at the same time: *Home Test Init* (HoTI) message via the HA and *Care-of Test Init* (CoTI) message directly. Upon the reception of each message, the CN sends back two messages to the MN: *Home Test* (HoT) message via the HA and *Care-of Test* (CoT) message directly, each containing a different token to be used by the MN to generate the binding management key. This binding management key is then used by the MN to send a verifiable BU to the CN.

A.3 Fast Handoffs for Mobile IPv6 (FMIPv6)

During handoffs in Mobile IP protocols there is a period during which the MN is unable to send or receive packets because of link switching delay and protocol operations. This handoff latency is often unacceptable to real-time traffic. FMIPv6 [71] was proposed to reduce the handoff latency of Mobile IPv6 and to prevent the quality of service degradation that an MN could suffer. Handoff processes are started when the MN is still present on the current link by using L2-triggers, which indicate that the MN will perform a handoff soon.

The idea behind this protocol is providing an MN with the IP subnet parameters to which it is going to move before it has actually done so. FMIPv6 also makes it possible to prevent packet loss through buffering and tunneling.

There are two modes in FMIPv6: predictive mode and reactive mode. Whenever possible, an MN will perform predictive handoffs as this type of handoffs will permit the MN to fully benefit from all FMIPv6 optimizations. The reactive handoff mode will be mainly used when a node has unexpectedly lost connection with its current AR. Predictive and reactive modes are depicted in Figures A.3 and A.4 respectively.

A.3.1 Predictive Handoff

After discovering nearby APs, through the *Router Solicitation for Proxy Advertisement* (RtSolPr) and the *Proxy Router Advertisement* (PrRtAdv) messages, the MN formulates a prospective new CoA (nCoA) when it is still present on the previous Access Router (pAR) link. If an MN is able to detect the need for a handoff through the use of L2 information, it sends a *Fast Binding Update* (F-BU) to its current pAR informing it to forward all the future packets to the new Access Router (nAR). This message contains the current CoA and the AR to which the MN is planning to switch. At that point the pAR sends to the nAR a *Handoff Initiate* (HI) message indicating the MN's link-layer address, the MN's previous CoA (pCoA) and the proposed nCoA, the nAR informs the pAR with a *Handoff Acknowledge* (Hack) message indicating whether the proposed nCoA is valid or not providing further nAR specific details. Upon Hack reception, pAR sends a *Fast Binding Acknowledgment* (F-BAck) to acknowledge receipt of a F-BU message. The MN receives it on the the pAR's link. This means that packet tunneling is already in progress by the time the MN handoffs to the nAR. The MN sends the *Unsolicited Neighbor Advertisement* (UNA) message immediately after attaching to the nAR. That allows the nAR to forward the stored packets to the MN right away, providing expedited forwarding of packets to the MN.

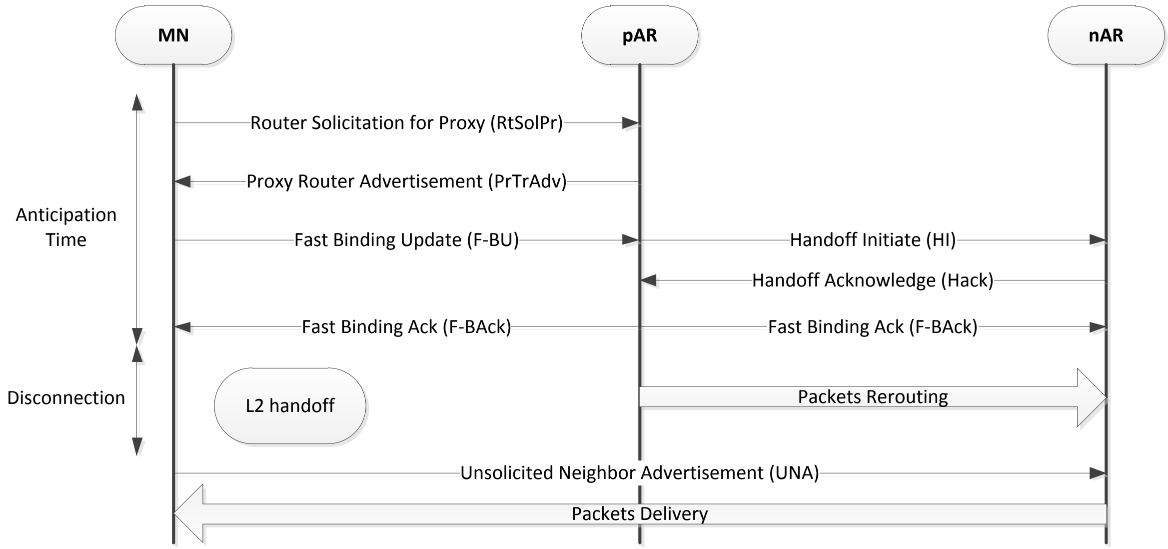

Fig. A.3 Predictive FMIPv6 Handoff

A.3.2 Reactive Handoff

This mode is used in case an MN is not able to anticipate a handoff and therefore is only able to react once it is already in progress, thus the disconnection with the pAR is suffered before the handoff process. A reactive handoff is produced when the MN leaves the previous link before sending the F-BU message or the message is lost or corrupted just before leaving the link. The MN can notice the situation because it will not receive the F-Back from the pAR. The MN will send the UNA message immediately after attaching to the nAR. The MN sends the F-BU message to the pAR immediately after sending the UNA message. The nAR then forwards that F-BU to the pAR, and the pAR starts tunneling packets. NAR responds to the UNA message in case it wishes to provide a different IP address to use. In reactive handoffs all packets sent between the disconnection of the MN from the pAR and the reception of the F-BU on the pAR are lost. PAR starts buffering packets to be rerouted after receiving the F-BU from the nAR.

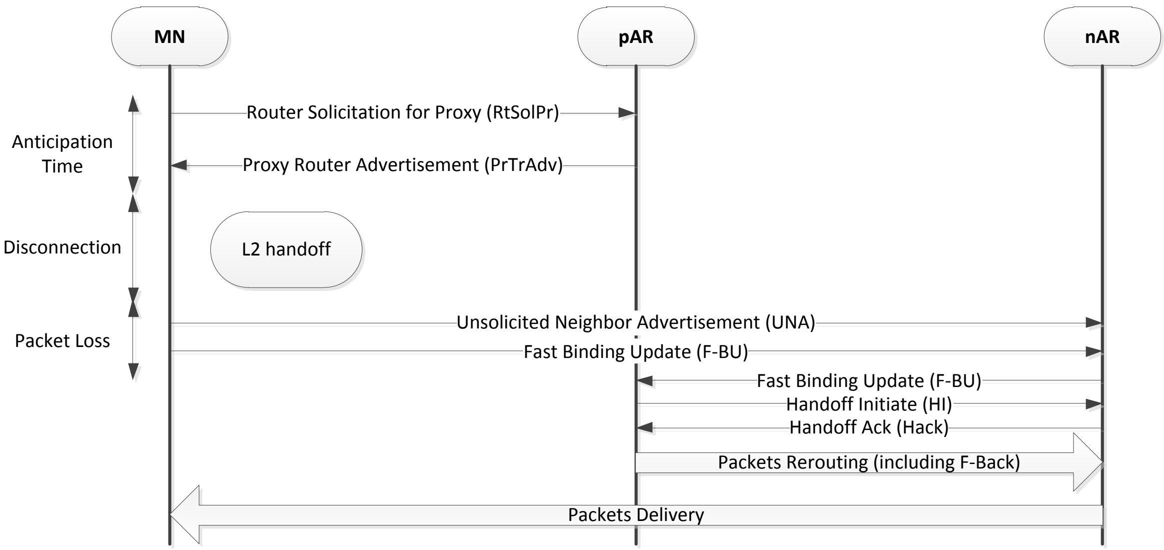

Fig. A.4 Reactive FMIPv6 Handoff

Appendix B

VESPA guidelines

VESPA and a set of video testing tools developed for the test-bed used in Chapter 4 can be freely downloaded from http://sourceforge.net/projects/vespa. There is also the possibility of downloading VESPA as a Live CD. The following sections of this document are aimed to explain how to build and install the VESPA platform. A section explaining the configuration of VESPA simulations is also included. At the end of the document, a section describing how to run a sample Demo Application is presented. The currently supported SUMO version is 0.12.3.

B.1 Downloading

The set of files required to install VESPA platform are available at http://sourceforge.net/projects/vespa/.

B.2 Building and Installing in Ubuntu 12.04

VESPA platform has been tested in Ubuntu 12.04.

B.2.1 SUMO

First of all we need to install SUMO traffic simulator version 0.12.3, available at http://sourceforge.net/projects/sumo/files/sumo/version0.12.3/sumo-src-0.12.3.tar.gz/download

- We need to install the following libraries.

  ```
  # apt-get install libxerces-c-dev libfox-1.6-dev
  ```

- We compile and install the source code

```
# tar zxvf sumo-src-0.12.3.tar.gz

# cd sumo-0.12.3

# ./configure

# make

# sudo make install
```

B.2.2 ns-2

In order to use VESPA platform we need the ns-2 version available at http://sourceforge.net/projects/vespa/files/ns-allinone-2.31-vepra.tar.gz/download

- We need to install the following libraries.

```
# apt-get install gcc-4.4 g++-4.4 libpcap-dev
```

- We need to compile the network simulator using gcc version 4.4

```
# export CC=/path_to_gcc-4.4

# export CXX=/path_to_g++-4.4
```

- We compile and install the source code

```
# tar zxvf ns-allinone-2.31-vepra.tar.gz

# cd ns-allinone-2.31-vepra

# ./install.sh
```

B.2.3 VESPA

Once we installed SUMO and ns-2 network simulator, in order to execute VESPA platform we need to download the platform files at http://sourceforge.net/projects/vespa/files/vepra.tar.gz/download

- To launch VESPA, first of all we need to install the following packages.

  ```
  # apt-get install vtun uml-utilities default-jre
  ```

- We decompress the downloaded file.

  ```
  # tar zxvf vepra.tar.gz

  # cd vepra
  ```

- We need to initialize TUN/TAP interfaces. In that case we enable 3 interfaces, one for each virtual machine network interface, and one to redirect the X server of the video client UML machine.

  ```
  # sudo ./virtnet init 3
  ```

- We configure the interface to redirect the X server of one virtual machine (in case is necessary).

  ```
  # sudo ./display
  ```

- We launch VESPA plaftorm.

  ```
  # java -jar vepra.jar
  ```

B.3 VESPA Simulation Configuration

VESPA platform provides a graphical user interface to facilitate the creation of vehicular scenarios to be tested. In Figure B.1 we can observe the GUI.

The GUI functionalities are distributed in the following way:

1. General Menu Button: This menu can be used to save and load previous configurations, and to set the path of ns-2.

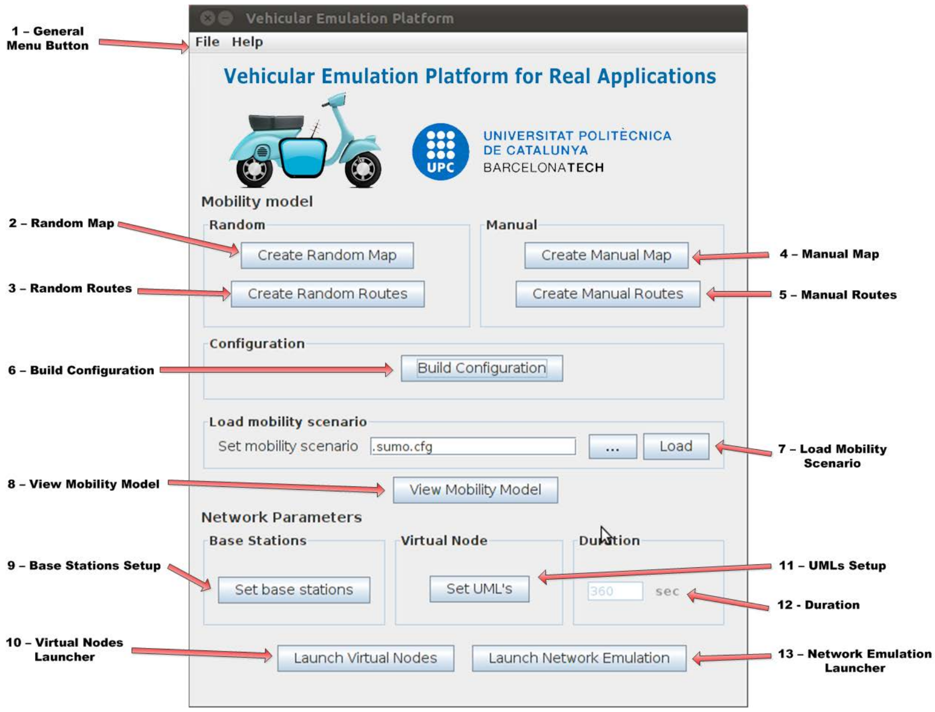

Fig. B.1 VESPA GUI

2. Random Map: Using this menu we can create a set of random maps using SUMO.
3. Random Routes: Using this menu we can generate traffic over the random maps that we created previously.
4. Manual Map: Disabled in this beta version.
5. Manual Routes: Disabled in this beta version.
6. Build Configuration: Used to generate config files used to load traffic scenarios.
7. Load Mobility Scenario: Using this menu we can load a config file of a vehicular scenario that we created using SUMO.
8. View Mobility Model: We can visualize the vehicular scenario loaded previously.
9. Base Stations Setup: We can configure the positions and access technology of the access points that we want to deploy in the scenario.
10. UMLs Setup: Using this menu we can configure the nodes that we want to virtualize using UML machines.
11. Duration: Time duration of the simulation.

12. Virtual Nodes Launcher: Once we configured the scenario we can use this button to boot up the virtual nodes.

13. Network Emulation Launcher: Once we configured the scenario we can use this button to start the network emulation.

B.3.1 Random Map Generator

This functionality of VESPA (Figure B.2) can be used to generate a random SUMO map. SUMO offers the possibility to generate three types of random maps:

- Grid: We can generate a Manhattan scenario, configuring the number of junctions, and the length of the streets.
- Spider: We can generate a scenario in a web style, configuring the number of axes, circles and the distance between circles.
- Random: Totally random scenario.

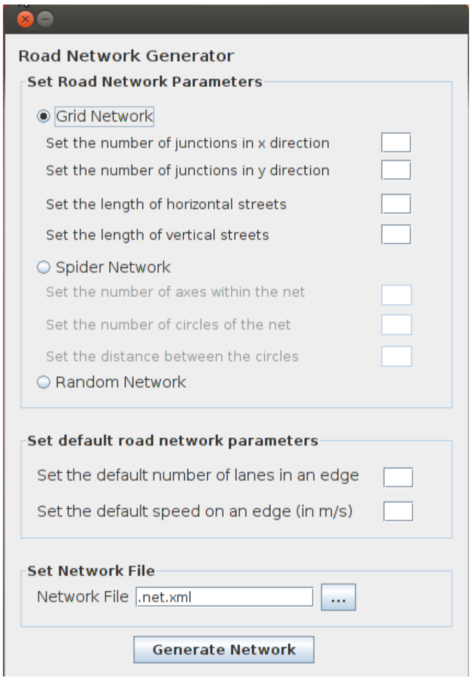

Fig. B.2 Road Network Generator

We need to fill the menu with the required parameters, and then we save the generated map in a file with the extension ".net.xml".

B.3.2 Random Traffic Generator

This menu (Figure B.3) can be used to generate random traffic in a random map previously created. To generate the random traffic we need to select the map file and set the generated random traffic in a file with the extension ".rou.xml".

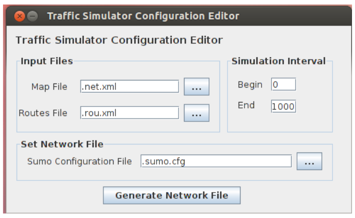

Fig. B.3 Random Traffic Generator

B.3.3 Configuration File Editor

This menu (Figure B.4) can be used to generate a configuration file. This configuration file is required by SUMO to load a vehicular scenario (map+traffic). In this menu we select both input files (map file and route file) and we set the config file that will be generated with the extension ".sumo.cfg".

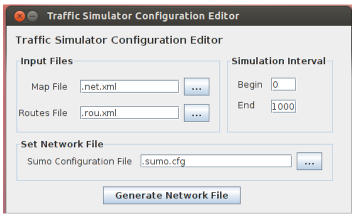

Fig. B.4 Configuration File Editor

B.3.4 Load Scenario

Once we created a vehicular scenario using SUMO, either using VESPA or not, we can load this scenario that will be used in the test-bed. We can see the scenario loaded below the "Load Mobility Scenario" form (see Figure B.5).

B.3.5 Network Infrastructure Configuration

Here we can configure the RSUs that will be deployed in the scenario. All these RSUs will be connected to a central server that will be virtualized via a UML machine, and the handoffs

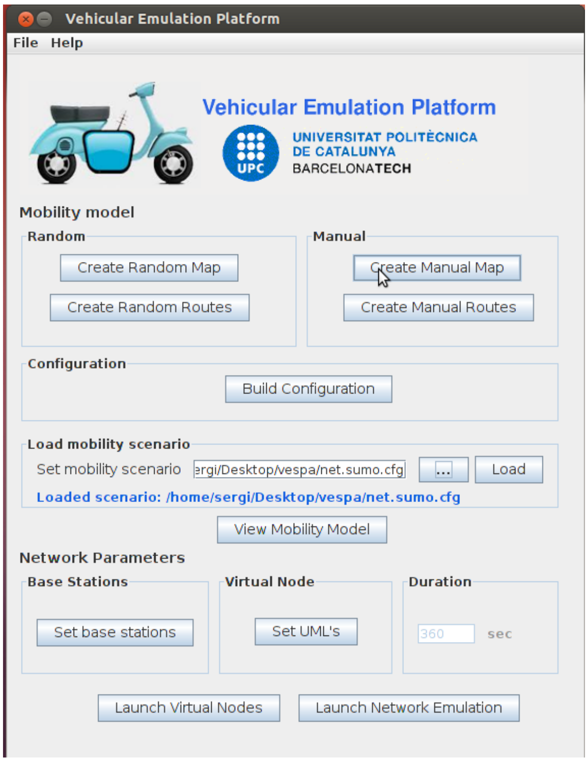

Fig. B.5 Load Scenario

between RSUs will be performed using Mobile IP. In the menu shown in Figure B.6, we need to configure the coordinates of the map where the RSUs will be placed, and the access technology (802.11p or 802.11b) used.

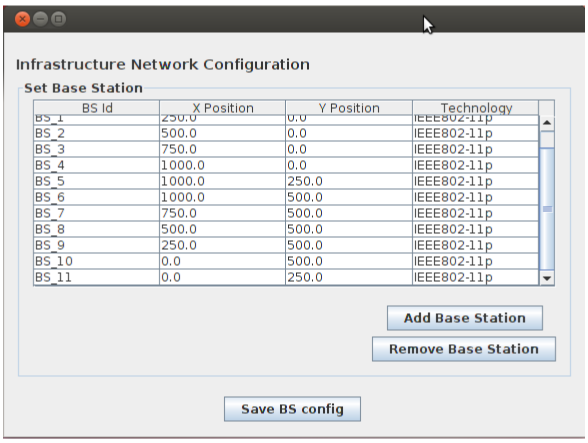

Fig. B.6 Network Infrastructure Configuration

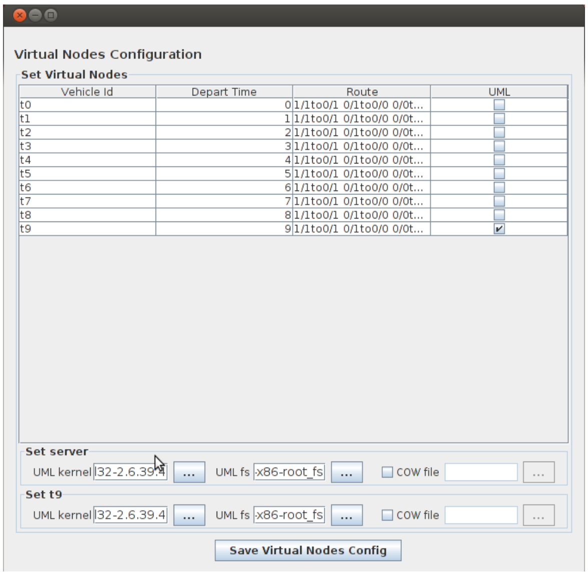

Fig. B.7 Virtual Nodes Configuration

B.3.6 Virtual Nodes Configuration

In this menu (Figure B.7) we must configure the virtual nodes. We can see a list of all the vehicular nodes configured in the vehicular scenario previously loaded. By default, a central server in the infrastructure is virtualized. So we can select any of the vehicular nodes in order to represent them by a virtual machine. Once we select a node, we have to set the UML kernel and filesystem that will be used for this virtual node. We can also use a *copy_on_write* file to be able to share the same filesystem between different nodes.

B.4 How To Run the Demo Application

If we download the Ubuntu Virtual Machine with VESPA installed, we can run a demo scenario. This scenario is the following:

In this scenario (Figure B.8), there is a simple highway vehicular scenario, with a vehicular node and a central server (CN). A set of RSUs (FAs) are deployed in the road, in an overlapped manner, so there aren't coverage blackouts. We will virtualize both, the vehicular node and the central server. In the central server we will use a UML virtual machine with a streaming server installed on it, and in the vehicular node we will use a video player. To run the demo scenario, we follow the guidelines below:

- First of all, we need to run VESPA. We open a terminal and we enter the following commands:

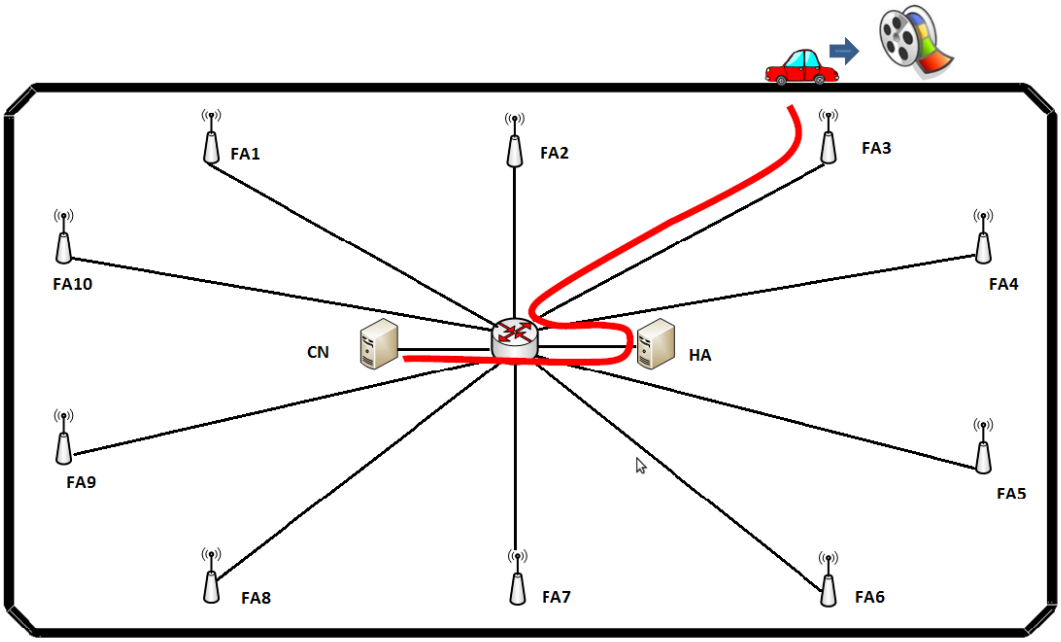

Fig. B.8 Demo Scenario

```
# cd Desktop/vespa/vepra

# sudo sh virtnet init 3

# sudo sh display

# java -jar vepra.jar
```

- Once we launched VESPA, we need to load the VESPA configuration already created. Menu File→Load Configuration (Figure B.9).

Fig. B.9 Load Menu

- Then we select the config file named *sample.sumo.cfg* in the folder */home/vespa/Desktop/vespa/vepra/samples* (Figure B.10).

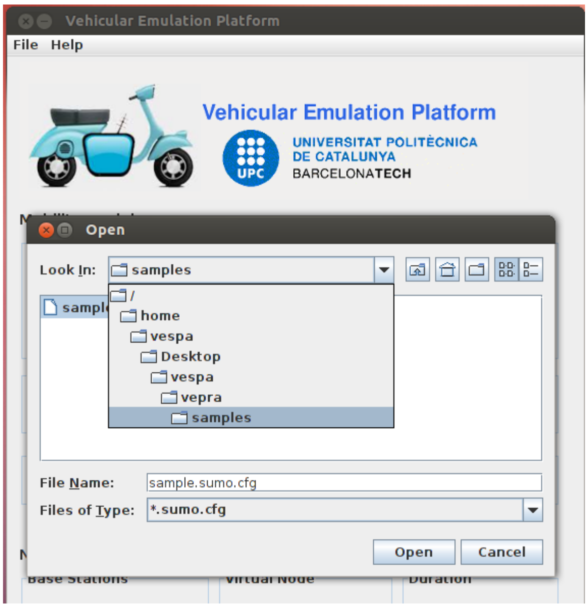

Fig. B.10 Load Configuration

- If we want to visualize the traffic scenario (optional), we can run *sumo-gui* application (Figure B.11). We need to load the scenario */home/vespa/Desktop/vespa/vepra/samples/sample.sumo.cfg* in the "Load Mobility Scenario" form, and we press the "Load" button. Then we press the "View Mobility Model" button.

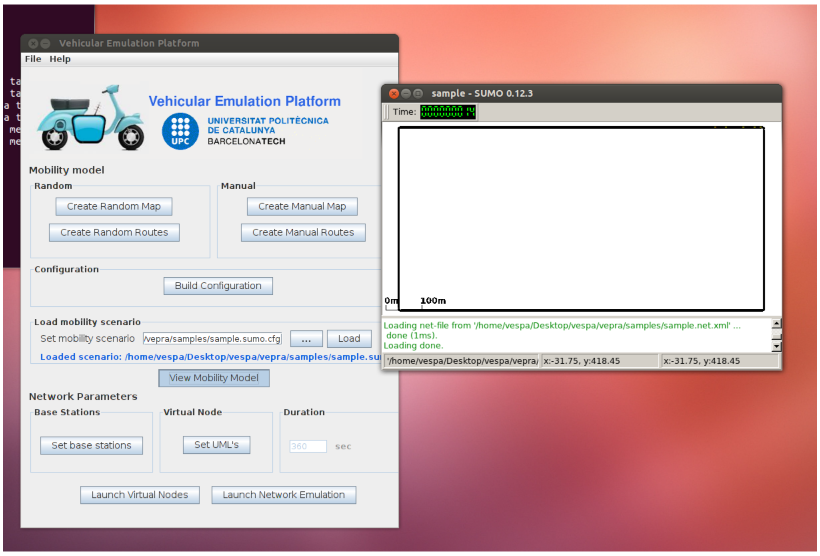

Fig. B.11 sumo-gui

- We need to run the UML virtual machines, using the “Launch Virtual Nodes” button, and then we run the network emulation using the “Launch Network Emulation” button. At this moment three terminals are opened, the two virtual machines, and a third terminal for the ns-2 console. Ns-2 requires the root password.
- When the virtual machines are booted up (Figure B.12), we can login using “root”. No passwords are required. In the server node, we run the video streaming server.

  `# sh start_video`

  In the vehicular node, we run the video player.

  `# sh video_client`

Fig. B.12 Demo Application

B.5 How to Run VESPA using command-line

We can also run VESPA using only the command-line. We need to launch the UML virtual machines and ns-2.

- Launch ns-2

  `# cd Desktop/vespa/vepra`

  `# sudo sh virtnet init 3`

```
# sudo sh display
```

```
# sudo ./ns-allinone-2.31-vepra/bin/nse emulation.tcl
```

- Launch UML server

```
# cd Desktop/vespa/vepra/uml
```

```
# ./kernel32-2.6.39.4 ubd0=Debian-server-x86-root_fs eth0=tuntap,tap0,FE:FD:00:00:00:00
mem=1024MB
```

- Launch UML node

```
# cd Desktop/vespa/vepra/uml
```

```
# ./kernel32-2.6.39.4 ubd0=Debian-vuser-x86-root_fs eth0=tuntap,tap1,FE:FD:00:00:00:01
eth1=tuntap,tap2 mem=1024MB
```