POLITECNICO DI TORINO

SCUOLA DI DOTTORATO

Dottorato in Ingegneria Elettronica e delle Comunicazioni – XXV ciclo

Tesi di Dottorato

Vehicular Networks with Infrastructure: Modeling, Simulation and Testbed

III Facoltà di Ingegneria
ING-INF/03

Massimo REINERI
Matr. 169452

Tutore
prof. Carla-Fabiana CHIASSERINI

Coordinatore del corso di dottorato
prof. Ivo MONTROSSET

Febbraio 2013

Summary

This thesis focuses on Vehicular Networks with Infrastructure. In the examined scenarios, vehicular nodes (e.g., cars, buses) can communicate with infrastructure roadside units (RSUs) providing continuous or intermittent coverage of an urban road topology.

Different aspects related to the design of new applications for Vehicular Networks are investigated through modeling, simulation and testing on real fields. In particular, the thesis:

i) provides a feasible multi-hop routing solution for maintaining connectivity among RSUs, forming the wireless mesh infrastructure, and moving vehicles;

ii) explains how to combine the UHF and the traditional 5-GHz bands to design and implement a new high-capacity high-efficiency Content Downloading using disjoint control and service channels;

iii) studies new RSUs deployment strategies for Content Dissemination and Downloading in urban and suburban scenarios with different vehicles mobility models and traffic densities;

iv) defines an optimization problem to minimize the average travel delay perceived by the drivers, spreading different traffic flows over the surface roads in a urban scenario;

v) exploits the concept of Nash equilibrium in the game-theory approach to efficiently guide electric vehicles drivers' towards the charging stations.

Moreover, the thesis emphasizes the importance of using realistic mobility models, as well as reasonable signal propagation models for vehicular networks. Simplistic assumptions drive to trivial mathematical analysis and shorter simulations, but they frequently produce misleading results. Thus, testing the proposed solutions in the real field and collecting measurements is a good way to double-check the correctness of our studies.

Acknowledgements

I would like to thank my supervisor Prof. Carla-Fabiana Chiasserini together with Prof. Claudio Casetti: it was a real pleasure to work so profitably with them. I hope our friendship and working collaboration will last forever.

I would like to thank also the whole Telecommunication Group at the Politecnico di Torino: there I found great friends and colleagues.

Furthermore, I am thankful to Prof. Mario Gerla and his research team at the University of California, Los Angeles: the period that I spent abroad in U.S.A. was really valuable, thanks also to the priceless support of the Fulbright program.

Finally, I thank my family for the love and trust they have placed in me over these years.

Contents

Summary II

Acknowledgements III

1 Introduction 1

2 Seamless Connectivity and Routing 4
2.1 Problem Statement 4
2.2 Related Work 6
2.3 Network System and Reference Scenarios 8
2.4 Routing in Vehicular Networks with Infrastructure 11
2.5 Layer-2 Implementation and Seamless Handover 18
2.6 Performance Evaluation in the Testbeds 20
2.7 Conclusion 26

3 Content Downloading Using UHF Band 28
3.1 Problem Statement 28
3.2 System Scenario 30
3.3 Protocol Description 31
3.4 Testbed Setup 32
3.5 Experimental Results 35
3.6 Conclusion 38

4 RSUs Deployment in Intelligent Transportation Systems 39
4.1 Problem Statement 40
4.2 Related Work 41
4.3 RSU Deployment for Content Dissemination and Downloading 43
4.4 Mobility Scenarios 47
4.5 Performance Analysis of the Heuristic under Ideal Network Settings . . . 48
4.6 Performance Analysis of Heuristic in Realistic Environments 55
4.7 Conclusion 59

5 Optimization of Urban Traffic Flows 61
5.1 Problem Statement . . . 61
5.2 Contribution . . . 62
5.3 Example Scenario . . . 63
5.4 Vehicles Behavior in a Urban Scenario . . . 64
5.5 Model . . . 66
5.6 Evaluation . . . 70
5.7 Related Work . . . 72
5.8 Conclusion . . . 75

6 Routing Strategies for Electric Vehicles 76
6.1 Problem Statement . . . 77
6.2 Related Work . . . 78
6.3 System Scenario and Problem Statement . . . 79
6.4 Why a Game Model? . . . 80
6.5 The Recharging Game . . . 82
6.6 Simulation Scenario . . . 87
6.7 Results . . . 88
6.8 Conclusions . . . 93

7 Conclusions and Future Work 94

Bibliography 96

Chapter 1

Introduction

In the last few years, the interest in Intelligent Transportation Systems (ITS) has been steadily increasing, fueled by the need for safety and entertainment applications. Roadways can be made safer by letting vehicular users communicate road and traffic conditions, as well as position and velocity. Also, since a person often spends in the car between one and two hours per day, most newly-manufactured vehicles boast multimedia capabilities, which beg for advanced infotainment services (email/social network access, newscasts, or local touristic clips).

To this aim, vehicular networks enable vehicles to communicate either with roadside units, in what is widely known as vehicle-to-infrastructure (V2I) communication, or among themselves, through vehicle-to-vehicle (V2V) communication. Several applications can be supported by either of the above communication paradigms, including transportation safety services, traffic monitoring and infotainment. While the former class of application requires broadcasting or geocasting of alarm or warning messages in a reliable, efficient manner, the latter two classes often imply the support of high-data rate, UDP-based traffic, such as video streams.

In this thesis we focus our attention in the case where vehicular nodes (e.g., cars, buses) can communicate with infrastructure roadside units providing continuous or intermittent coverage of an urban road topology, depending on the application we are going to study. In particular, the thesis covers problems related to routing, connectivity, Road Side Units (RSUs) placement and applications for Vehicular Networks. Different scenarios have been studied through mathematical analysis, simulations and measurements on the fields using real prototypes.

In Chapter 2, we consider a scenario where the RSUs provide continuous coverage to the vehicles traveling along the roads. There, the wireless mesh network represents the infrastructure and the vehicular nodes are mesh nodes themselves. Mesh nodes connect over the wireless medium and act as routers, and data packets may traverse multiple wireless hops. In that case, our aim is to design a solution that guarantees low jitter and high packet delivery ratio to sustain high-data rate UDP-based applications for the vehicular

users, such as video streaming. The problem is first addressed through simulations, then the solution is validated in two real testbeds.

In Chapter 3, we reuse the system architecture described before, but we address the problem of content downloading exploiting the benefit of using UHF bands for the transmission of control messages, so as to make it more efficient. In fact, in previous studies we realized that the 5 GHz bands offer limited capacity channels in comparison to the broad range of services envisioned in vehicular networks. Indeed, we design a protocol for content downloading that leverages the UHF band for control messages and the high-throughput, 5-GHz bands for data delivery. In this scenario, the RSUs using 5-GHz bands provide intermittent coverage (like hot-spots), while the LRU (Long Range Unit) using UHF band provides continuous coverage of the area. The results have been validate through a testbed and compared with respect to the case where only 5-GHz bands are used.

Also in Chapter 4, we study content downloading, along with content dissemination, but there we address the problem of RSUs deployment to ensure good performance to bypassing users. The RSUs deployment implies costs related to the hardware, installation and power consumption that are proportional to the number of installed devices. Indeed, over-dimensioning the network installing a big number of RSUs could guarantee high performance but it is not affordable for the operators' point of view due to its cost. Thus, we propose new RSUs deployment strategies to find the right trade-off between costs and performance, providing intermittent connectivity of the urban area but still guaranteeing good service quality for both dissemination and content downloading.

In Chapters 5 and 6, we envision two applications that exploit the Vehicular Network to provide real services to the users. The peculiarity of these chapters is that they address problems related to routing of vehicles instead of routing of data packets. Functionally, we assume to have a system architecture allowing the exchange of information among vehicles, for instance like one of those presented in the previous chapters. In the considered scenarios, each vehicle gathers the required traffic information towards a Central Controller using the available infrastructure, and relies on common navigation services. In Chapter 5, we propose a method to optimize urban traffic layout using basic heuristics and computationally efficient simulations. Instead of modeling an entire urban map with hundreds of intersections, each typology of intersection is simulated in order to understand how it responds to different traffic patterns and intensities. Then, this knowledge is leveraged to allow the computation of minimal delay route on the complete road map.

In Chapter 6, we address the problem of Electric Vehicle drivers' assistance through ITS. Drivers of EVs that are low in battery may ask a navigation service for advice on which charging station to use and which route to take. A rational driver will follow the received advice, provided there is no alternative choice that lets the driver reach its destination in a shorter time, i.e., in game-theory terms, if such advice corresponds to a Nash-equilibrium strategy. Therefore, we solve the problem using a game-theoretic approach, envisioning two models, namely a congestion game and a game with congestion-averse

utilities, both admitting at least one pure-strategy Nash equilibrium. Using our models, we show that the average per-EV trip time yielded by the Nash equilibria is very close to the one attained by solving a centralized optimization problem that minimizes such a quantity.

Finally, in Chapter 7, we present the conclusions that we draw from this work.

Chapter 2

Seamless Connectivity and Routing

In this chapter, we consider vehicles (e.g., cars, buses or streetcars) that connect to different roadside mesh nodes as they move in an urban environment, and we study the joint problem of traffic delivery and connectivity management in such scenario. In that case, our aim is to support high-data rate UDP-based applications, such as video streaming, designing a solution able to guarantee low jitter and high packet delivery ratio. The problem is first addressed through simulations, then the solution is implemented and tested in two real testbeds.

The content of this chapter is organized as follows. Section 2.1 describes in details the requirements to support high-data rate UDP-based services in Vehicular Networks with Infrastructure. In Section 2.2 we review previous work, while in Section 2.3 we describe our network system and we present two reference scenarios, which will be used for performance assessment. Section 2.4 addresses the problem of efficiently routing a traffic connection between vehicles and infrastructure, and shows that our proposal based on the BATMAN protocol gives excellent performance. Section 2.5 introduces the channel selection scheme and the seamless handover procedure we designed. Finally, Section 2.6 presents the performance results obtained through two different testbeds, and Section 2.7 draws some conclusions.

2.1 Problem Statement

We consider a scenario of high practical relevance, in which the network infrastructure is represented by a wireless mesh network and the vehicular node is a mesh node itself. Mesh networks are typically free-standing, robust systems that can be conveniently integrated with the existing infrastructure and offer high bit-rate services. Mesh nodes, also called *mesh points*, connect over the wireless medium and act as routers, and data packets may traverse multiple wireless hops. Note that, while the literature already features works that address the mobility of user devices [1–4], they are usually seen as end nodes

rather than mesh routers. We specifically address the case in which the vehicular node is a mesh router. The advantage of our solution is twofold: (i) it allows the routing protocol to be run on the mobile node itself, thus better adapting to the high-mobility profile of the node; (ii) the node on the vehicle becomes a full-blown "mobile hot spot" that can act as a gateway towards the mesh infrastructure for all client devices on board the vehicle. Also, as often done in mesh and vehicular networks [5–8], we envision several frequency channels to be available for V2I communication as well as for communication between roadside units, and that more than one radio interface is available at both the roadside and the vehicular mesh points.

In such a scenario, uninterrupted connectivity is nominally guaranteed, but a rapidly-reacting routing protocol is needed to handle sudden link quality drops caused by mobility or channel fading; instrumental to ensuring connectivity and performance is therefore the degree of responsiveness of the routing protocol. Also, a seamless procedure that allows vehicles to identify the "best" channel available and hands over the connection from one infrastructure mesh point to another is needed to guarantee sustained end-to-end throughput, as well as low jitter (e.g., suitable for multimedia streaming to/from a mobile node), throughout the journey of the vehicles.

As discussed more extensively in Section 2.2, several works have dealt with reliability of V2I communication links, channel access and support of QoS at the MAC layer for real-time services, while few studies have considered the problem of V2I connection management and seamless handover of UDP-based streams. Even fewer works have explored these issues through experimental measurements in realistic network scenarios. As for the literature on wireless mesh networks, again, several studies have focused on interference and frequency allocation in multichannel systems, or on routing in totally fixed or totally mobile mesh networks, while the joint problem of routing and mobility support in a scenario with vehicular and roadside mesh points has been scarcely addressed.

In our work, we define an on-board fast-switching layer-2 architecture for 802.11-based mesh networks with mobility support, which allows vehicles to efficiently communicate with the wireless mesh infrastructure. Generally, layer-2 routing for mesh networks has been proposed in the 802.11s draft standard [9], and, even earlier, within the IETF MANET Working Group [10]. We identify a scheme that features implementations both at the layer 2 and at layer 3, i.e., the BATMAN [11] protocol, which has been developed by the Freifunk Mesh Community and is becoming increasingly popular among developers. We assess the performance of BATMAN by comparing it with three routing protocols, each of which represents a different approach to routing: (i) AODV [12], a reactive scheme which has inspired the Hybrid Wireless Mesh Protocol specified in the IEEE 802.11s Draft Standard [9], (ii) OLSR [13] and OLSR-ETX, well-known proactive routing protocols, (iii) GPSR [14], a geographical routing protocol. To avoid full-scale, time-consuming experiments, which would have been difficult to carry out, we use ns2 simulations and run these schemes at layer 3 of the mesh nodes in the scenario outlined above. We find that, when the traffic flows from a vehicle towards the infrastructure

(hereinafter referred to as uplink), the best performance is achieved by both GPSR and BATMAN; the latter, however, fails to provide good results when traffic flows in the opposite direction (hereinafter referred to as downlink). By investigating this contradictory behavior, we identify a problem of the BATMAN protocol related to its window-based mechanism for path quality estimation, and we solve it. We name the improved version of the protocol as smart-window (sw-) BATMAN, and, through further simulations, we observe that it yields good performance, both in uplink and downlink.

Based on these findings, we select sw-BATMAN for routing traffic, and we deploy two real vehicular testbeds where sw-BATMAN is implemented at layer 2. Such an implementation choice on the one hand allows the increase of its operations speed, on the other it simplifies the network configuration, not requiring IP addresses to be assigned to mesh points. We then devise and implement a channel selection scheme that lets a vehicle connect to the infrastructure by using always the "best-quality" link, and we design a handover mechanism that allows vehicles to connect to the roadside mesh node through the channel used by the best-quality link. This enables a seamless transfer of data as vehicles move, thus resulting in excellent performance in terms of throughput and delay jitter.

2.2 Related Work

Prior studies using IEEE 802.11 radio technology have evaluated the feasibility of both V2I and V2V communications, in a real word scenario. In particular, the works in [15,16] present measurements taken with cars running at different speeds and show that the main factor affecting V2I connectivity is the distance between vehicle and infrastructure. Such result is confirmed by the study in [17], where again the performance of V2I communications is shown to depend on communication range and line of sight.

At the MAC layer, several solutions have been proposed for the support of real-time services, although most of them focus on safety applications. Examples are the proposals in [18] and [19–21]. The scheme in [18] is designed to provide QoS support in vehicular Internet access: it employs fixed gateways along the road that perform periodic admission control and scheduling decisions for the packet traffic in their service area. The works in [19–21], instead, aim at defining a fast connection setup mechanism for V2I communication, when vehicles are equipped with one radio interface only and the 802.11p random access protocol is used. The authors envision a centralized, polling-based access scheme running at each roadside unit, on top of 802.11p, so as to provide an upper bound to the delay experienced by safety traffic. Furthermore, each vehicle sends information on its position, speed and direction so that roadside gateways can predict when a vehicle enters their service area and promptly include it in their traffic schedule. Analytical and simulation results show the limited overhead of the solution and the feasibility of safety-critical V2I applications in a dense-traffic, highway scenario.

At higher protocol layers, the problem of V2I connectivity has been addressed via simulation, e.g., in [22], where a hybrid architecture, called Vehicle-to-Vehicle-to-Infrastructure (V2V2I), is evaluated. In V2V2I, the transportation network is broken into zones in which a single vehicle, namely the Super Vehicle, is the only one able to communicate with the infrastructure. All other vehicles can only communicate with the Super Vehicle. This solution reduces the contention among vehicles to access a single roadside unit, although the Super Vehicle-Infrastructure link may represent a bottleneck.

Well-known solutions for mobility support at the network layer are provided by the IETF activities, e.g., Mobile IP and NEMO [23]. The former allows host mobility without connection disruption, by letting a node acquire a care-of address and by redirecting traffic towards it; the latter instead enables entire IPv6 networks (i.e., a mobile router aboard a vehicle and a number of devices deployed within the vehicle) to change their point of attachment to the Internet [24].

We highlight that our handover mechanism, unlike Mobile IP or NEMO, works at layer 2; in this way, connectivity management is implemented through efficient, fast operations that make the support of UDP traffic, such as video streaming, possible. Also, since our solution is based on the layer-2 implementation of the BATMAN routing protocol, it leaves the MAC layer unchanged.

Finally, we point out that, while several works on mesh networks have focused on interference and frequency allocation in multichannel systems (e.g., [6, 7] just to name works based on testbeds), or on routing in totally fixed [25–30] or totally mobile wireless mesh networks [31], few papers have addressed the joint problem of routing and mobility support in a V2I scenario. Also, existing experimental works on wireless mesh networks with mobility support, such as [1–4], have considered neither mobile mesh nodes, nor the problem of ensuring a seamless procedure to let a highly-mobile mesh node connect to different fixed nodes as it moves. In particular, in [2] traffic disruptions during the handover of a mobile terminal between static mesh points are avoided thanks to the formation of a multicast group, which the mesh points currently serving the user have to join. The problem of applying such solution to our scenario lies on the overhead and on the multicast group management in a highly-mobile environment. In [3], again, a low-mobile terminal user is considered and a quick handover procedure in 802.11-based mesh networks is envisioned, by letting a single-interface user quickly scan the access points in range and choose the best, provided all nodes are synchronized. Besides the modifications to the required 802.11 driver, this mechanism does not address the problem of seamless handover, which is critical for the support of multimedia streams in a vehicular environment. Similarly to our work, the study in [4] considers a multi-channel, multi-radio terminal user that slowly moves between different 802.11 access points. Fast handover is provided by letting one of the radio interfaces working in scan mode while the other transmits/receives traffic. Such a solution, however, works at the MAC layer and does not address traffic routing.

Our study instead focuses on the management of vehicular-to-infrastructure connectivity, when the vehicle is a mesh point itself, and we jointly address traffic routing and handovers for the support of multimedia streams. Note that, in our case, the interfaces of all network nodes operate in ad hoc mode, thus we do not face the problem of association with access points, which is one of the main issues addressed in the literature on fast handovers in 802.11-based networks. We also assess the performance of our solution in a real setting, showing that bandwidth-demanding applications, such as those based on UDP streaming, can be successfully supported by a wireless mesh infrastructure, as vehicular mesh points move and change point of attachment to the fixed nodes. A preliminary version of our work has appeared in [32], where only a single V2I scenario was considered and a much simpler, less efficient handover scheme was presented.

2.3 Network System and Reference Scenarios

We focus on a mesh network consisting of several roadside mesh points and one mobile mesh node installed on a vehicle. The latter moves within the coverage of roadside mesh points (hereinafter simply called roadside points). Our objective is to devise a fast-switching layer-2 protocol architecture that can provide seamless, sustained-quality transmission of multimedia and data streams in both uplink and downlink directions, between a terminal endpoint attached to the vehicular mesh node and another endpoint reachable through one of the roadside points.

Assuming that the vehicle equipped with the mesh node travels along a route, with continuous coverage by roadside points, makes our testbeds amenable to being implemented on a public transportation line (buses, streetcars, low-speed leisure trains, etc.). In our setup we did not optimize transmissions for a mobile setting, i.e., by using variable-aperture or variable-gain antennas; rather, we tried to devise solutions to cope with handovers between different roadside points when the vehicular node is equipped with two radio interfaces. The use of two radio transceivers in vehicular networks has been considered in several papers [33, 34] and it receives nods also in the VANET standardization/industrial community [35, 36]. Finally, we point out that our testbed scenarios deliberately feature areas with suboptimal coverage quality. Thus, the behavior of our architecture under adverse (though realistic) conditions could be investigated.

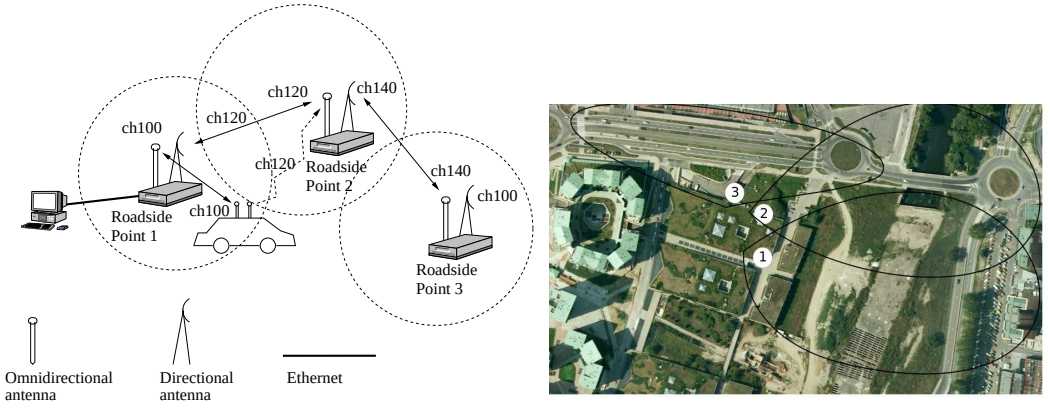
The Hardware and Software Platforms

All hardware devices use off-the-shelf components and run Open Source software. The network nodes in the testbeds are installed at the roadside and on board a car that was modified to accommodate external antennas. The nodes are enclosed in water-resistant small-size boxes (180 mm$\times$125 mm$\times$46 mm), thus allowing ease of installation. Each node is fitted with an Alix PC Engines motherboard, equipped with an AMD Geode 500

MHz processor; additionally, each has a compact flash memory of 1 Giga Byte, an Ethernet card and two Ubiquiti Networks XtremeRange 5 (5.5-5.7 GHz) IEEE 802.11 radio cards. Radio cards are compliant with the 802.11h specifications on spectrum and transmission power management, but they support the data rate and MAC specified by the 802.11a standard. The set of available channels that we consider is composed of 11 channels (from channel 100 to channel 140), each 20 MHz-wide. The radio cards driver is MadWifi revision 3314, with OpenWRT patches.

Description of the Testbeds

To analyze the behavior of the overall system, two main testbeds were created. The first one is based on three roadside points that cover a densely traveled-by urban road and its aim is to evaluate the communication infrastructure in a real environment. The second testbed was created on a private road, using several roadside points, where the aim was the assessment of handovers and multi-hop communications.



(a) Abstract representation

(b) Real view: locations of the roadside points are marked by circles

Figure 2.1: Network scenario in the first testbed

The first testbed is set up on a 1-km stretch of public road, in a busy urban area that provides plenty of obstacles (both fixed and moving), thus matching the expected real operating conditions. Three roadside points set 50 m apart cover the stretch of road from a vantage point, and a car equipped with a mesh node is driven between the two ends of the road segment, trying to maintain a constant 36 km/h speed, which was not always possible due to existing traffic (indeed, the actual average speed turned out to be nearly 18 km/h). A continuous UDP stream (either in uplink or in downlink) is arranged between a laptop carried on the car and a desktop reachable through the infrastructure nodes. The network topology, as far as the roadside points are concerned, has a linear structure, as shown in

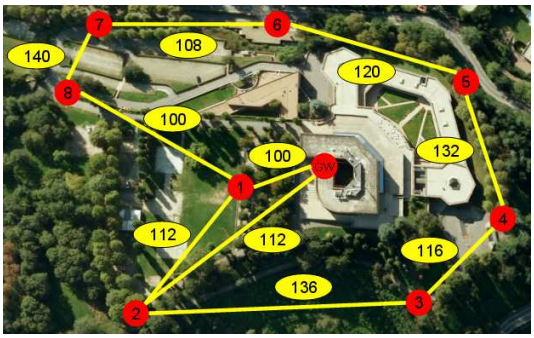


Figure 2.2: Network scenario in the second testbed. Circles represent the eight roadside points and the gateway node, while continuous lines represent the wireless links between roadside points and with the gateway. The numbers within the ovals (namely, 100, 108, etc.) denote the channels used on the links

the abstract representation of the reference scenario in Figure 2.1(a). The deployment of the three roadside points is shown in Figure 2.1(b).

Each roadside point has two external antennas: one of them is omnidirectional, with a gain of 9 dBi and transmit output power of 8 dBm, while the other exhibits a limited aperture (either 18 degrees with 18 dBi gain and transmit output power of 10 dBm, or 60 degrees with 16 dBi gain and output power of 9 dBm[1]). In particular, the highly directional antenna is used to establish a link with the omnidirectional antenna of the next roadside point along the path. The omnidirectional antenna, beside forming one end of the link with the previous roadside point in the path, also captures the transmissions of the vehicle, when in range. Different, non-adjacent frequency channels are chosen for each antenna and for the antennas of neighboring nodes, in order to limit interference through frequency diversity [37]. A sample choice of channels is shown in Figure 2.1(a), where channels 100, 120 and 140 are used and are such that the central frequencies of channels 100 and 120, as well as of 120 and 140, are 100 MHz apart. The vehicle is instead equipped with two omnidirectional antennas, each with gain of 9 dBi and transmit output power equal to 6 dBm.

The second testbed is set up on a 1-km loop on a private road in a woodland area; the full coverage is guaranteed by 8 roadside points and outside connectivity is achieved through another mesh point, acting as a gateway, installed on the roof of a nearby building. Figure 2.2 describes the position of the roadside points and the channels that were assigned to each link to avoid interference. Roadside points, as well as the car (which is driven at the average speed of 18 km/h) are equipped with two omnidirectional antennas,

[1]Note that the values of transmit output power that we used are much lower than the maximum value allowed by regulation, i.e., 1 W.

with a gain of 9 dBi and transmit output power of 18 dBm[2].

2.4 Routing in Vehicular Networks with Infrastructure

We identify BATMAN as a candidate layer-2 implementation of a routing protocol. As already mentioned, layer-2 routing has been already pursued in mesh and vehicular networks. Before proceeding with the implementation of BATMAN in a real testbed, we use simulation to compare its performance with other routing protocols for wireless ad hoc and mesh networks. Since most of them only feature layer-3 implementations, we also consider the layer-3 BATMAN version, which retains all the mechanisms of its layer-2 counterpart.

Next, we briefly recall the main features of the protocols that we consider and show the results derived through simulation that resemble the settings used in our testbeds. BATMAN shows some inconsistencies in its behavior that require a closer look at its mechanisms. After the problem is identified, we propose a solution and evaluate its impact, before committing ourselves to the final architectural choice for our testbeds.

The Protocols under Study

Since we do not aim at a comprehensive comparison of the performance of *all* routing protocols for wireless ad hoc and mesh networks known in the literature, we choose just a few that represent a cross-section of a broader collection of protocols. Specifically, we include both reactive and proactive routing protocols, as well as protocols that use either a link-state, a distance-vector, or a geographic approach. Below, we present the main features of each scheme, highlighting the version of the protocols that we implemented.

AODV: it builds routes on demand by flooding the network with route request messages. As a route request hits the intended destination or a node that is aware of a fresh route towards the destination, the request is not forwarded any further and a reply message is sent back to the source. While traveling through the network, route request and reply messages create paths pointing, respectively, to the source and to the destination. Sequence numbers, route error messages, and local repair are used to handle link failures and avoid loops. Variations of the protocol [38, 39] give a node the ability to store more than one route per destination. However, since in our highly-dynamic scenario such enhancement provides little improvement in performance, we stick to the protocol version specified in the RFC 3561 [12].

OLSR: it adopts a proactive, optimized link-state scheme to spread topology information while keeping the overhead low. The key idea is that link-state information is generated

[2]The presence of several trees and other obstacles suggested the use of a higher power than in the first testbed.

and flooded in the network only by selected nodes, called Multi Point Relays (MPRs). Any source-destination route is bidirectional and includes only MPRs as relay nodes. In this work, we refer to the protocol specified in the RFC 3626 [13], which selects the shortest path between a source and a destination. However, we also consider the version that includes the Expected Transmission Count (ETX) metric [29], so that routes with higher packet delivery ratio can be selected.

GPSR: it exploits the nodes' geographical position to make packet forwarding decisions. For each packet, the source polls a lookup service to acquire the location of the intended destination, then a greedy approach is used to forward packets to nodes that are progressively closer to the destination. When no greedy path exists, GPSR forwards traffic in *perimeter* mode. In this case, a packet traverses successively closer faces of a planar subgraph of the full radio network connectivity graph until, reaching a node closer to the destination, the greedy forwarding resumes. We refer to the protocol version described in [14].

BATMAN: it is a proactive protocol based on the distance-vector paradigm, therefore its strategy is to determine, for each destination in the network, the neighbor that can be used as best next hop towards the destination. To learn about the best next hop for each destination, all nodes periodically broadcast originator messages (OGMs) to their neighbors; each OGM contains an *originator address*, a *sending node address* and a *unique sequence number*. When a neighbor receives an OGM, it changes the sending address to its own address and rebroadcasts the message if either the OGM was originated by a neighboring node or the OGM was received from a node that is considered the best next hop towards the originator. To identify the best next hop towards a destination, a node counts the number of OGMs originated by the destination and received from the different neighbors. The node records this information in the so-called originator list. Then, it selects as next hop the neighbor from which it has received the highest number of OGMs within a sliding window (packet count metric), i.e., the path with best quality. In the following, we will refer to the packet count metric as *path quality level*. In this way, a node does not maintain the full route to a destination but every node on the path only knows the next hop to use to reach it. Note that, in the originator list, for each destination every node maintains as many sliding windows as the number of neighbors from which the node has been receiving OGMs originated by that destination. Furthermore, a node removes a neighbor from its originator list if it does not receive any OGM for a given time; changes in the originator list may lead to a routing table update. In this work, we refer to the protocol description in [11].

Below, we compare the above protocols by using ns2 simulations, in a network scenario similar to the one depicted in Figure 3.1. The simulation scenario, however, includes eight roadside points; also, the number of vehicles travelling along the road as well as

their speed are varying system parameters. We then use a setting that matches the description of the testbeds in Section 2.3, e.g., we set the node radio range[3] to 250 m while, to represent the different quality level of the links between roadside points and vehicle, and of the links between roadside points, we set the maximum data rate of the two types of links to 6 and 54 Mb/s, respectively. Both the cases of uplink and downlink CBR traffic are investigated. For all protocols, the control messages used to assess the connectivity with neighboring nodes (i.e., Hello messages in AODV and OLSR, Beacon messages in GPSR, and OGMs in BATMAN) are periodically transmitted with time interval equal to 1 s and, being broadcast messages, they are sent at the basic rate of 6 Mb/s. In GPSR, we assume a vehicle always has a perfect, instantaneous knowledge of the position of the destination through an ideal location service (which is not simulated). The information on the neighbors' position, acquired through the GPS and included in the GPSR beacon messages, is also considered to be error-free.

We remark that all curves have been derived by averaging the results over 10 different-seed runs, obtaining a confidence level of 95%; also, the plots show the confidence interval, which is represented through error bars.

Figure 2.3 presents the average received throughput at the application layer as the bit rate of the CBR traffic varies, in uplink (left plot) and downlink (right plot). The results refer to the case with one vehicle travelling at a constant speed of 36 km/h. In BATMAN, the sliding window size is set to 128. We observe that, in uplink, BATMAN and GPSR achieve the best performance. Indeed, BATMAN sends OGMs on every interface and collects statistics on the quality of all existing paths, thus promptly selecting the route that minimizes packet loss and delay latency. Similarly, in GPSR, the next-hop selection, which exploits the information included in the beacons to maximize the advancement of traffic towards the destination, allows a quick reaction to topology changes.

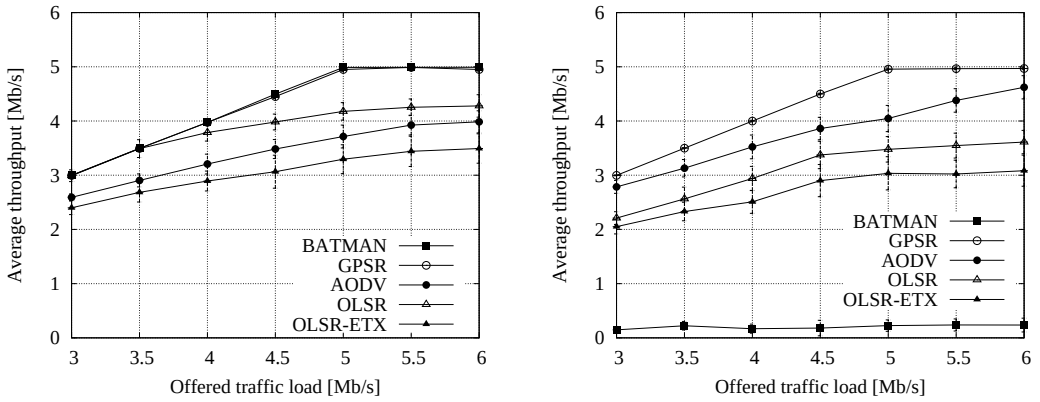
As for AODV, we observe that whenever the link between the source (i.e., the vehicle) and its next hop fails due to mobility, a new, fully-fledged route discovery is started, thus leading to performance degradation. We remark that, unlike BATMAN, which constantly monitors the quality of all paths through OGMs and switches promptly upon performance degradation, AODV only acts upon compromised connectivity with the previous next-hop and ongoing packet loss.

We now look at the results given by OLSR, which, disregarding link quality, provides a lower throughput than BATMAN as the offered load increases. OLSR-ETX does account for the link data rates, however, as shown in [40], the ETX metric takes quite a long time to detect a link failure, thus leading to worse performance than OLSR in a dynamic network. Finally, we point out that the better performance of BATMAN comes at the

[3]The node radio range in the testbed was measured as the maximum distance from a transmitter at which a node can receive traffic with a packet error rate smaller or equal to 0.08 (as typically considered in 802.11 networks).

cost of a higher message overhead; nevertheless, the overhead due to BATMAN is still negligible compared to the system capacity (the ratio is indeed of the order of 10^{-3}).

Next, we look at the performance in presence of downlink traffic. OLSR and OLSR-ETX give slightly lower values of throughput than in the case of uplink traffic. Indeed, due to their low reactivity, OLSR and OLSR-ETX fail in providing the source with an updated link-state information so as to route traffic correctly towards the destination (which is now moving). As for the other schemes, GPSR still provides a high throughput, consistently with the results for the uplink transfer. AODV too yields good performance, indeed, when a link breaks, the upstream node operates a local repair to recover connectivity to the destination, thus avoiding a new route discovery. BATMAN, instead, surprisingly yields the worst performance. In the next section, we investigate the behavior of BATMAN and try to find a solution to the observed performance degradation.



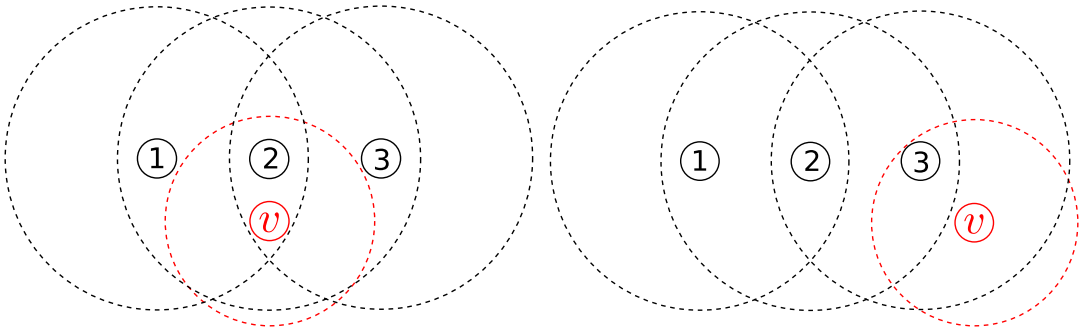
(a) Uplink (b) Downlink

Figure 2.3: Protocols comparison in terms of average throughput, as the offered traffic load varies and for a vehicle speed of 36 km/s

Identifying and Solving the Problem in BATMAN

BATMAN counts the number of OGMs within a sliding window to choose the best next-hop node towards a given destination. In the BATMAN sliding window, all OGMs have the same weight, i.e., older and newer messages have the same importance. By analyzing the protocol behavior, we noticed that in the scenario under study (i.e., a vehicular mesh node connecting to roadside mesh nodes) such window management may cause temporary routing loops and, thus, packet losses.

As an example, consider a network at time t_0, as shown in Figure 2.4(a): nodes 1, 2 and 3 are roadside points, while node v is mobile (circles with dashed lines represent the node radio range). Using a window size of 128 packets and an OGM interval time of 1 s [11], at t_0 the BATMAN routing tables at the four nodes are as in Table 2.1.



(a) t_0 (b) t_1

Figure 2.4: Network topology at different time instants

Table 2.1: Routing tables at time t_0

Node 1		
Dest.	Next hop	Quality
2	2	128
3	2	128
v	2	128

Node 2		
Dest.	Next hop	Quality
1	1	128
3	3	128
v	v	128

Node 3		
Dest.	Next hop	Quality
1	2	128
2	2	128
v	2	128

Node v		
Dest.	Next hop	Quality
1	2	128
2	2	128
3	2	128

Observe that node 1 does not receive OGMs directly from v and, thus, it uses node 2 to reach v; also, all routes report the maximum value[4] of the path quality level (i.e., the packet count metric). Now, assume that v reaches the new configuration shown in Figure 2.4(b) at time t_1. The routing tables at the four nodes at t_1 are as in Table 2.2. Note that, within a timeout, set by default to twice the OGM interval, a node purges from its originator list a neighbor from which OGMs are no longer received. Thus, nodes 2 and v realize that their link failed and update their routing tables accordingly. On the contrary, node 3 starts receiving OGMs directly from v, however, since BATMAN does not discriminate between recent and older events, at t_1 node 3 still maintains node 2 as next hop towards v. It follows that v can successfully send data to any other node, while losses may arise in the opposite direction. For instance, if node 1 wants to transmit a

[4]In BATMAN finding the best route towards the destination corresponds to finding the best hop towards it; the quality level metric at a generic node is therefore given by the number of OGMs originated by the destination and received from a neighbor (candidate next-hop) within a sliding window.

Table 2.2: Routing tables at time t_1

Node 1		
Dest.	Next hop	Quality
2	2	128
3	2	128
v	2	125

Node 2		
Dest.	Next hop	Quality
1	1	128
3	3	128
v	3	11

Node 3		
Dest.	Next hop	Quality
1	2	128
2	2	128
v	2	117

Node v		
Dest.	Next hop	Quality
1	3	7
2	3	7
3	3	7

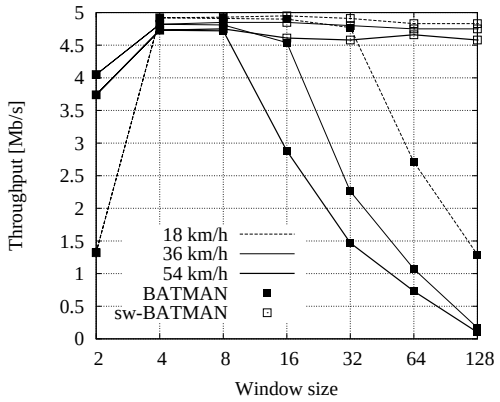


Figure 2.5: Comparison between BATMAN and sw-BATMAN: average throughput in downlink, as sliding window size varies, and for different vehicle speeds

packet to v, it sends it to node 2, which forwards the packet to node 3. Node 3 checks its routing table and sends the packet back to node 2; the packet is then bounced between node 2 and node 3 till its Time To Live expires and the packet is dropped.

It is therefore clear that the problem lies in the use of the sliding window, which slows down the reactivity of the protocol in the face of topology changes. A possible solution would be to decrease the window size, but it would trade off reactivity with route flapping, and the choice of the ideal size of the sliding window would still depend on the node speed. This can be clearly seen by looking at Figure 2.5, which shows the downlink throughput provided by BATMAN as the size of the sliding window varies. These results refer to the case of a single vehicle generating CBR traffic at 6 Mb/s and travelling at different speeds.

We therefore take a different approach and modify the sliding window mechanism, so as to tangibly reduce the problem without increasing the protocol complexity or altering

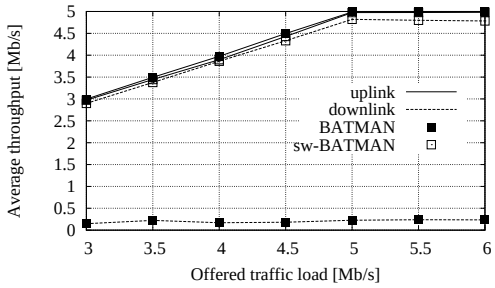


Figure 2.6: Comparison between BATMAN and sw-BATMAN: average throughput in uplink and downlink, as the offered traffic load varies

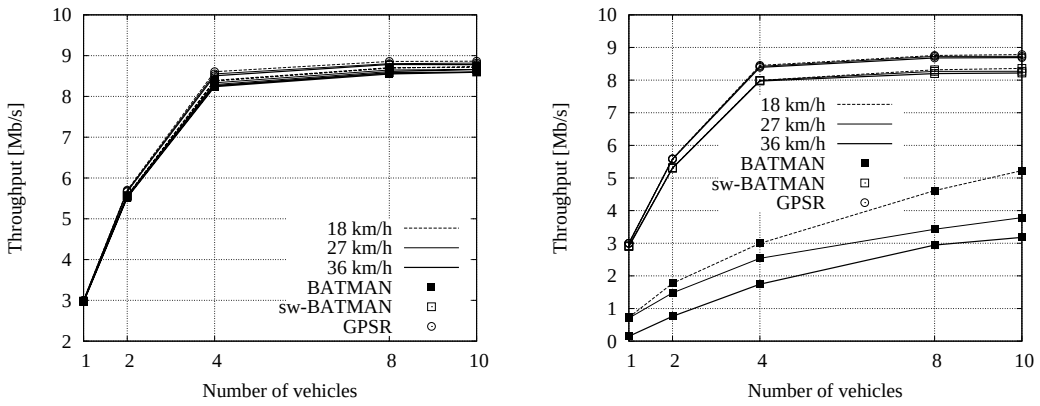
the spirit of BATMAN.

Our solution consists in changing the way OGMs are counted in the sliding window so that newer OGMs, representing fresh information, are weighted more than older OGMs. The weights, however, must be chosen in such a way as to avoid fluctuations in the routing tables: if the weight of fresher OGMs is too high compared to the others, the reception of OGMs from different neighbors with similar link quality may lead to a continuous change of the next-hop towards the destination. In our study, we considered the following expression for the values of the weight vector,

$$w(i) = \max\left(1, \left\lfloor \frac{i \cdot S}{2^i} \right\rfloor\right) \quad i = 1, \ldots, S \tag{2.1}$$

where $w(i)$ is the i-th element of the weight vector and S is the sliding window size (namely, 128); $w(1)$ is associated to the freshest OGM, older OGMs being recorded for increasing values of i. Note that the expression in (2.1) is a good choice because (i) it implies an exponential decrease of at least the more recent values (this is a standard practice when dealing with time windows implementing exponential averaging, i.e., to bestow greater importance to recent values than to older ones), (ii) it can be easily implemented in the kernel of a communication node, (iii) it does not include excessively large weight values so as to avoid overflow. We name this modified version of the protocol as smart window (sw-) BATMAN.

By looking at Figure 2.6 and comparing the results (obtained under the same scenario and settings as before) with the curves in Figure 2.3, we notice that sw-BATMAN provides excellent results in both uplink and downlink. Such good performance is confirmed by the results in Figure 2.7, which shows the performance of BATMAN, sw-BATMAN and GPSR (in uplink as well as in downlink), as the number of vehicles and their speed (namely, 18, 27 and 36 km/h) vary. At the beginning of these simulations, half vehicles move in one direction and the other half in the opposite direction. The offered load corresponding to each traffic transfer is equal to 3 Mb/s; the sliding window size of BATMAN and sw-BATMAN is still set to 128. Furthermore, Figure 2.5 shows that, unlike



(a) Uplink (b) Downlink

Figure 2.7: Comparison between different BATMAN versions and GPSR, as the number of vehicles varies and for different vehicle speeds

BATMAN, the performance of sw-BATMAN only marginally depends on the choice of the sliding window size (except for a window of size 2, for which route flapping is inevitable). While the exponentially decreasing weights used by sw-BATMAN consistently provide the reactivity needed to achieve a sustained throughput under any window size, the same cannot be said of standard BATMAN, where the "good" values of window size depend on the vehicle speed.

Finally, we remark that the overhead due to sw-BATMAN is just slightly higher (few kb/s higher) than in BATMAN, due to its quicker reaction to topology changes: the number of generated OGM packets is the same in both versions of the protocol, but the number of rebroadcasted OGMs is higher in sw-BATMAN. Indeed, OGMs are rebroadcasted by a node only if they have been originated by a neighbor, or if they have been received via a bidirectional link that currently serves as the best link towards the originator. Thus, upon a topology change, in BATMAN some OGMs are not rebroadcasted because they are received over a link that is (wrongfully, having the best link changed) not considered the best.

2.5 Layer-2 Implementation and Seamless Handover

The forwarding protocol used in our testbeds is layer-2 BATMAN Advanced Kernel Land, modified in order to include the smart-window mechanism. Being a layer-2 implementation, all OGMs are encapsulated in their own Ethernet frames and the whole testbed

network can be seen as a giant Ethernet switch[5]. Thanks to the layer-2 implementation, BATMAN runs in kernel space, rather than in user space, increasing speed and responsiveness of the protocol. BATMAN Advanced Kernel Land uses MAC addresses as routing table identifiers, instead of IP addresses, which also eliminates the need to assign IP addresses to mesh points.

A software module was added to the vehicular mesh point so as to handle the crucial tasks of selecting the "best-quality" channel available and of achieving a seamless handover between two roadside points. Recall that, in order to avoid interference, roadside points are assumed to provide their coverage using a frequency channel that is different from the one used by nearby roadside points. The solution we devise (i) manages the vehicle's radio interfaces in order to establish layer-2 connectivity to nearby roadside points whenever possible, and (ii) hinges on sw-BATMAN routing to trigger the handover.

More specifically, our solution leverages the following two factors: the availability of multiple radio interfaces at the vehicle and the fact that sw-BATMAN uses all available radios, each tuned to a different frequency channel. For clarity, below we will restrict the description to the case where two interfaces are available at the vehicle. Each vehicle radio interface is independently managed to identify a sw-BATMAN-capable neighbor, by intercepting its periodically transmitted beacon and reading the Basic Service Set Identifier (BSSID)[6]. The beacon also informs (in one of its Information Elements, IEs) whether the interface belongs to a roadside point or to another vehicle, in order to avoid V2V communication. If the BSSID is recognized as belonging to the vehicular network and the node interface issuing the beacon is part of the roadside infrastructure, the vehicle interface establishes an ad hoc link with it. OGM messages then start to be exchanged and statistics on the path quality are collected.

As the vehicle moves on, the quality of the link with the next hop along the current path is compromised. At the same time, the second interface will have established a link with the upcoming roadside point, and started to exchange OGMs through it. Conceivably, a better next hop (hence, path) will be known to sw-BATMAN through the second interface. sw-BATMAN thus switches the packets onto the new path and a "soft", seamless handover is performed[7]. If the path quality level associated with the channel used by the first interface has dropped below a given threshold T_q (set to the 80% of the maximum value of the path quality metric in our testbeds), the link with the previous roadside point interface is torn down and the MAC address of the roadside point interface is blacklisted. Blacklisting of a roadside point interface is introduced in order to temporarily (20 s in

[5]If the number of nodes scaled up in the tens or even hundreds, standard traffic-separation techniques at layer 2 (i.e., the superimposing of VLANs) could be used to dampen the ill effects of growing traffic.

[6]The BSSID is the same for all nodes interfaces in the vehicular network.

[7]If the vehicle had only one interface, the fact that adjacent roadside points use different channels would instead trigger a frequency scanning by the vehicle and, thus, an intermittent connectivity during handovers.

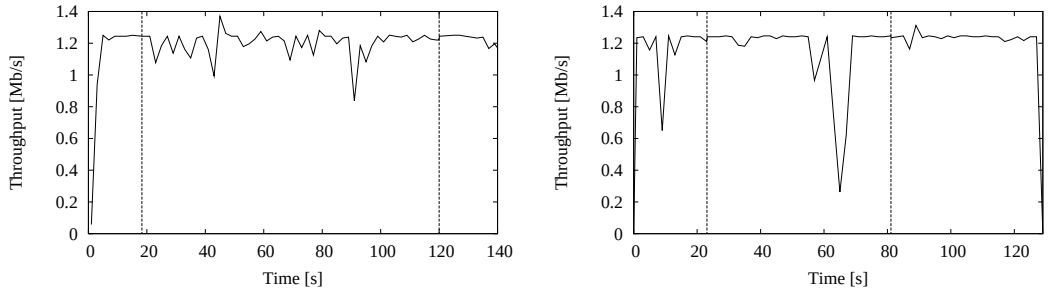
our testbeds) prevent the re-establishment of a recently torn-down link, thus avoiding a ping-pong effect. The 20 s duration we chose worked fine for our testbed, but it does not necessarily apply to generic testbed conditions. Finding an adaptive solution would be the optimal course, but we did not address this issue in the present work and reserve it for future study.

We can detail the interface management at the vehicular mesh point through a simple state machine that runs independently for each radio interface. The state machine can be in one of the following states: 1) scanning (the interface is trying to detect a roadside point), 2) connecting (the interface has detected a roadside point and is exchanging sw-BATMAN OGMs with it), and 3) active (the interface can be used to reach the next-hop found through sw-BATMAN routing).

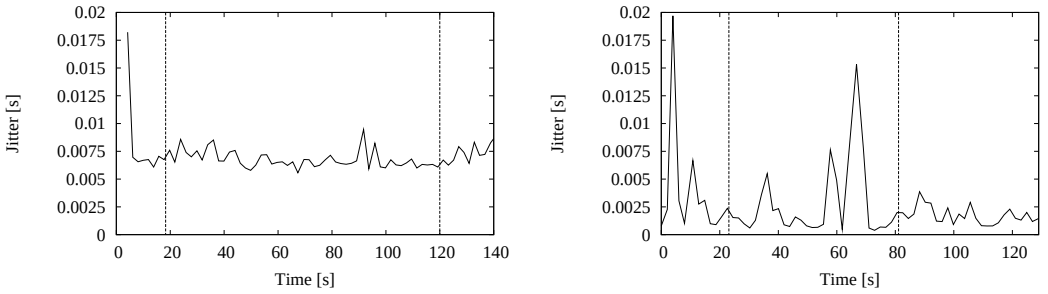
The main actions performed in the three states by the interface are described below.
1) Scanning state: in this state, the interface performs a channel scan to detect a roadside point and select a frequency channel. Note that the scanning interface skips the channel used by the active interface, if there is any. If more than one roadside point is available, the selection is based on the level of received signal power and on the MAC address blacklist: the channel with the highest received signal strength index (RSSI), measured on the beacon messages, and used by a non-blacklisted interface is chosen. The interface then moves to connecting state.
2) Connecting state: the interface starts collecting OGMs and monitors the trend of the path quality level every second. If the quality level does not decrease over time, the interface remains in connecting state until the path quality level becomes greater than the threshold T_q, and then it moves to the active state. Otherwise, the state reverts back to scanning.
3) Active state: the routing table entries associated to the interface are used by sw-BATMAN to pick the next hop for the outgoing traffic from the vehicular node; when both interfaces are active, sw-BATMAN ends up using the interface with best path quality to the destination (since this is the metric reported in each entry). Also, the node inspects the routing table associated with the interface every second, in order to verify the current path quality. If both interfaces are active, the vehicle node checks if the path quality level associated with the channel used by an interface falls below the threshold T_q; if so, such interface is moved to the scanning state and the roadside point interface is blacklisted. If only one interface is active, the threshold T_q is not considered and a state change of the active interface occurs only when the path quality level associated with the channel used by the interface reaches zero.

2.6 Performance Evaluation in the Testbeds

As explained above, in both testbeds we considered a CBR stream (either in uplink or in downlink) at 1.2 Mb/s, carried by UDP, with packet size equal to 1440 bytes to avoid



(a) Uplink (b) Downlink



(c) Uplink (d) Downlink

Figure 2.8: Temporal evolution of the throughput and delay jitter, as the vehicle moves from Roadside Point 1 to Roadside Point 2 and Roadside Point 3, in the first testbed

fragmentation. In the following, all throughput are measured at the MAC layer of the receiver (using TCPdump).

With regard to the first testbed, we observed the performance achieved as the vehicle moved from Roadside Point 1 to Roadside Point 2 and Roadside Point 3 (see Figure 3.1). As an example, Figures 2.8(a) and 2.8(b) show the throughput obtained during two of the observed vehicle trips: the former plot refers to a trip of duration equal to 140 s and with uplink traffic, while the latter refers to a trip of duration equal to 130 s and with downlink traffic. Figures 2.8(c) and 2.8(d) present the delay jitter, in uplink and downlink, observed during the same two trips. In the plots, vertical dashed lines indicate the time instants at which a handover takes place. Note that, in this case, a handover corresponds to a change in both channel and roadside point through which the traffic flows.

We observe that a seamless handover is performed at each transition from one roadside point to another: no throughput or jitter degradation occur due to the change of point of attachment. Low values of throughput and high jitter are instead experienced at certain time instants (e.g., at the beginning of the measurement period) when no line-of-sight communication is possible due to the presence of buildings along the road. Note also that bursts of delayed packets are responsible for throughput spikes in excess of 1.2 Mb/s (i.e., higher than the offered load), which immediately follow dips in throughput. Comparing uplink and downlink results, we observe that the behavior in downlink is smoother, except

for few negative spikes of short duration. The reason for the different throughput profiles lies in the fact that uplink transmissions occur at lower power than downlink transmissions. Consistently with these results, in the uplink transfer we note a higher jitter than recorded in downlink.

As for packet losses, in uplink we lost two and one packets in correspondence of the first and second handover, respectively, while in downlink three and four packets were lost during the two transitions. Similar performances were obtained in all the measurements we carried out.

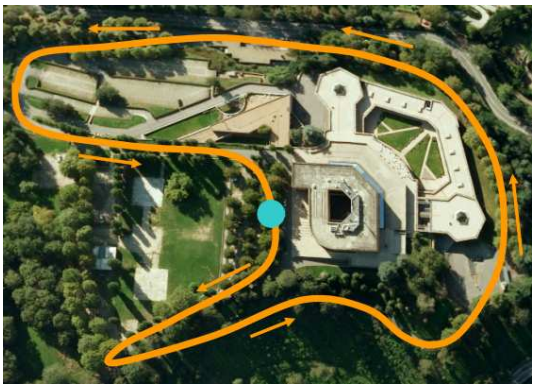


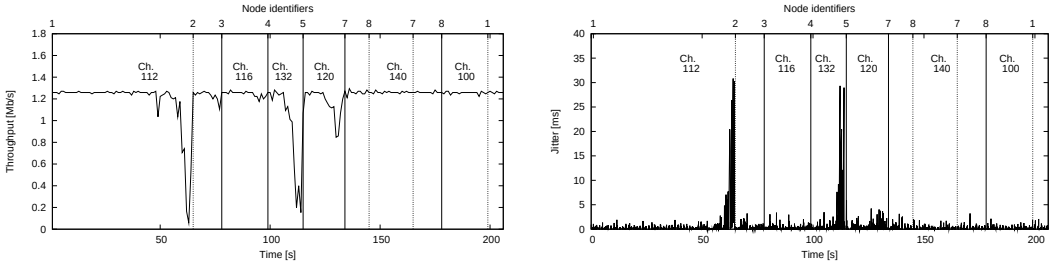
Figure 2.9: Route followed by the vehicle in the second testbed. The circle denotes the starting point of each lap

Next, we evaluate the system performance in the second testbed, using either one or two vehicles.

In the case of uplink and downlink traffic, the UDP flow is generated, respectively, by the vehicles and by the gateway node (GW in Figure 2.2) that connects the mesh network to the fixed infrastructure. Each reported test is sample single lap out of several back-to-back laps around the loop road, starting from the point tagged by the circle in Figure 2.9. Although we tried to drive at similar speeds throughout each lap, this was not always possible; different speeds, leading to different channel conditions, as well as slightly different trajectories are thus responsible for inhomogeneities in results referring to the same stretch of road. Throughput and jitter measurements for each lap are reported in Figures 2.10 through to 2.17; solid vertical lines represent transitions between different channels, while dashed vertical lines represent transitions between different roadside points on the same channel.

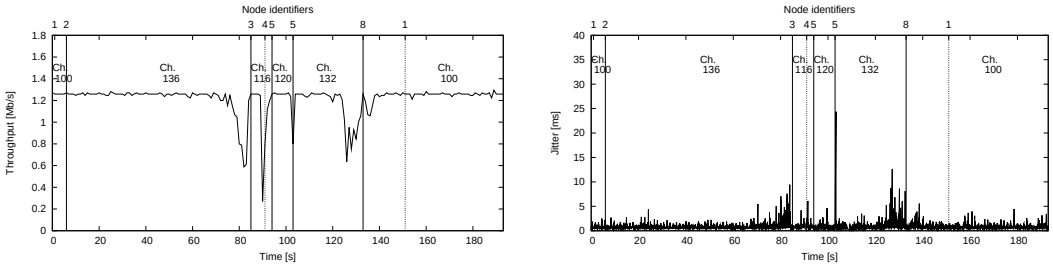
We first look at the uplink results in Figures 2.10 and 2.11, which refer to a faster and a slower lap, respectively. Both exhibit a qualitatively similar behavior in terms of throughput and jitter, although the faster lap features a slightly larger number of inter-nodal handovers (an intra-nodal handover, instead, can be seen in Figure 2.11, around $t = 103$ s).

In Figures 2.12 and 2.13, we present the throughput and jitter results, respectively,



(a) (b)

Figure 2.10: Temporal evolution of the throughput and delay jitter in uplink, in the second testbed (Test 1 - single vehicle)

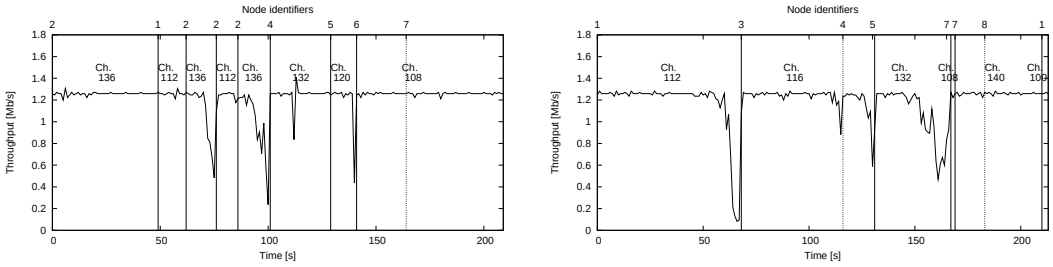


(a) (b)

Figure 2.11: Temporal evolution of the throughput and delay jitter in uplink, in the second testbed (Test 2 - single vehicle)

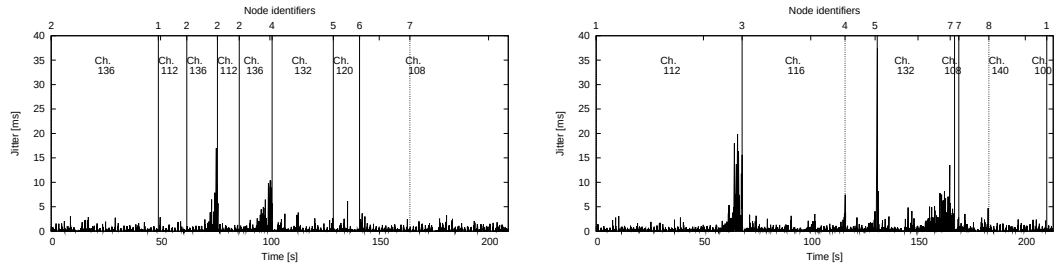
obtained by a pool of two vehicles both engaged in an uplink transfer. Interestingly, no visible performance degradation can be noticed, mainly due to the fact that the combination of channel selection and of sw-BATMAN routing lets each node use the least congested links.

Next, we shift our focus onto the downlink performance, shown in Figures 2.14 and



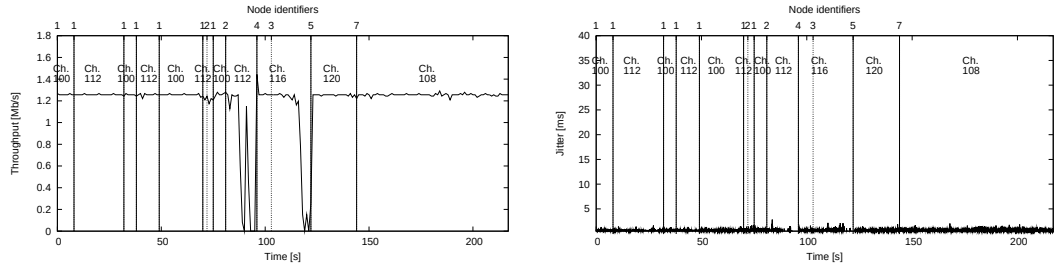
(a) (b)

Figure 2.12: Temporal evolution of the throughput in uplink, in the second testbed (Test 3 - two vehicles)



(a) (b)

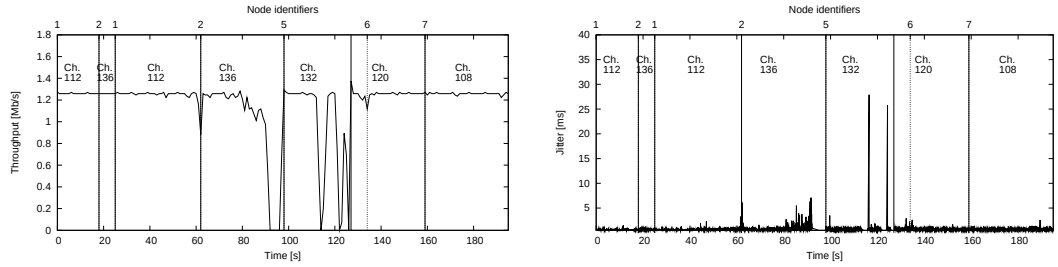
Figure 2.13: Temporal evolution of the delay jitter in uplink, in the second testbed (Test 3 - two vehicles)



(a) (b)

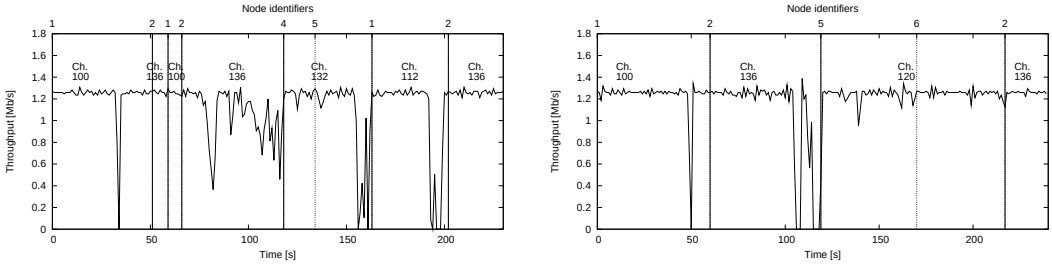
Figure 2.14: Temporal evolution of the throughput and delay jitter in downlink, in the second testbed (Test 1)

2.15, again referring to a faster and a slower lap, respectively. The degradation of throughput performance with respect to the uplink is quite evident. Indeed, as already remarked in previous sections of this chapter, sw-BATMAN is less reactive when the mobile node is the traffic destination. This implies that traffic is routed towards a roadside point that has already lost connectivity with the vehicle. Thus, the throughput exhibits short-lived "black-outs" as opposed to the low (but still positive) dips experienced in uplink. As an



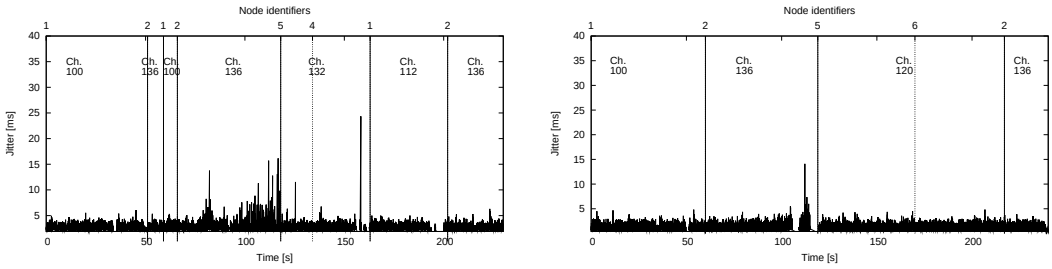
(a) (b)

Figure 2.15: Temporal evolution of the throughput and delay jitter in downlink, in the second testbed (Test 2)



(a) (b)

Figure 2.16: Temporal evolution of the throughput in downlink, in the second testbed (Test 3 - two vehicles)



(a) (b)

Figure 2.17: Temporal evolution of the delay jitter in downlink, in the second testbed (Test 3 - two vehicles)

example, considering the 5-second intervals before and after the channel change, in the first uplink test 1 (Figure 2.10(a)), the number of lost packets amounts to: 26 at $t = 78$ s, 9 at $t = 99$ s, 356 at $t = 115$ s, and 100 at $t = 134$ s. Instead, in the first downlink test (Figure 2.14(a)), in correspondence of the "black-outs" at $t = 96$ and $t = 122$ we have, respectively, 410 and 525 lost packets. This behavior however translates into lower jitter (i.e., packets are lost rather than accumulating variable delays due to MAC-layer retransmissions). Also, we point out that the brief node/channel handover seen around $t =$ 70 s (node sequence 2-1-2) may seem to contradict the blacklisting mechanism, while, in fact, it is due to both links' quality being above T_q, hence being both active. A similar phenomenon can also be seen in Figure 2.16(a), which shows the throughput experienced by one car of the pool, again in downlink. Finally, as in the uplink case, the results for the two-car pool are similar to the single-car case, except for a higher jitter (see Figures 2.16 and 2.17).

Finally, in Table 2.3 we present the average and the variance of the throughput, as well as of the percentage of packet losses during a channel change. The results have been obtained by averaging 39 tests with uplink traffic and 30 tests with downlink traffic (the plots not included in this chapter can be found in [41]). Looking at the table, we can see that very good results are obtained in both the uplink and downlink cases, although uplink transfers exhibit slightly better performance.

Figure 2.18: Snapshot from on-board camera streamed video in the second testbed

A subjective evaluation of the uplink transmission quality in the second testbed is provided by a video, available at [42]. The video was captured by an on-board-camera and then streamed from the vehicle to a server reachable through the GW node, where it was recorded. A snapshot of the video can be seen in Figure 2.18. The vehicle started its lap from the point tagged by the blue circle in Figure 2.9 and, in about 240 s, completed a full lap of the closed circuit. The video is somewhat garbled at 55 s (i.e., at the first handover); also, at 88 s, the video freezes for a few seconds during the second handover. On the whole, however, the quality of the uplink video transmission is remarkably good, also in critical situations, e.g., when the vehicle makes a U-turn.

In light of these results, we can conclude that our solution is capable of providing sustained end-to-end throughput as well as low jitter, throughout the journey of a vehicle along the path. Moreover, it guarantees a low packet loss even in critical environments, e.g., where the radio path between vehicles and roadside points is obstructed by trees and constructions.

2.7 Conclusion

We addressed the problem of seamless connectivity and channel selection between vehicular and roadside mesh points, a topic that is usually overlooked, since studies in the literature are more concerned with support for nomadic or slowly moving end users. We

Table 2.3: Throughput and lost packets per channel change

	Throughput [Mb/s]		Lost packets [%]	
	Average	Variance	Average	Variance
Uplink	1.17	0.049	6.6	3.3
Downlink	1.14	0.053	8.6	3.9

identified BATMAN as a possible layer-2 solution that could suit our aims, and, through simulation, we compared it with the most common routing protocols for ad hoc networks in a vehicular scenario with roadside infrastructure. From this comparison, some inconsistencies in the behavior of BATMAN emerged, and we proposed a solution to enhance its reactivity. After introducing a weighting mechanism in the window-based path quality estimation used by BATMAN, we implemented it in our testbeds, along with a channel selection mechanism and a seamless handover procedure. The performance observed in the two roadside vehicular testbeds proved the feasibility of our solution, and opened interesting perspectives in the use of mesh networks for the support of UDP-based services to vehicular users.

Chapter 3

Content Downloading Using UHF Band

In this chapter, we consider a system architecture made by different RSUs and similar to the one described in Chapter 2. Since in previous studies we realized that the 5 GHz bands offer limited capacity channels in comparison to the broad range of service envisioned in vehicular networks, here we investigate the benefit of using UHF band to extend the available system bandwidth. In particular, we design a new protocol for content downloading that leverages the large-coverage UHF band for control messages, and the high-throughput 5-GHz bands for data delivery. The efficacy of the proposed solution is proved through a testbed and the results have been compared with respect to the case where only 5-GHz bands are used.

The content of this chapter is organized as follows. Section 3.1 introduces the Content Downloading problem and reviews previous work, while Section 3.2 describes the network scenario that has been implemented in our vehicular testbed. The protocol message exchange for content downloading, on both the UHF and the 5-GHz bands, is introduced in Section 3.3. Section 3.4 details the testbed set up, while the results derived from our measurement campaign are presented in Section 3.5. Section 3.6 concludes the chapter highlighting directions of future research.

3.1 Problem Statement

In order to support advanced infotainment services and applications (email/social network access, newscasts, or local touristic clips), frequency spectrum regulations have licensed 5.9 GHz band or dedicated short-range communication (DSRC) for ITS, while the IEEE 802.11p specifications have standardized vehicular communications over the allocated spectrum. In particular, 802.11p foresees a time division technique to let a vehicle equipped with one radio operate on the control and service channels. Also, it allocates one frequency channel for control message exchange and safety applications, and six channels for other services, all of them in the 5.9 GHz band.

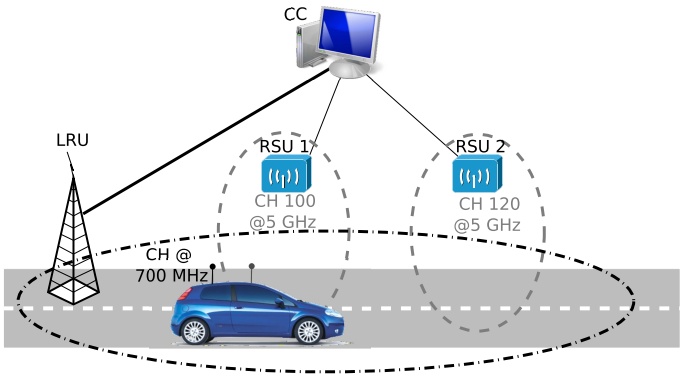


Figure 3.1: Abstract representation of the network scenario in the testbed.

Several research studies [43–47] suggest that, under high vehicle density or emergency situations, this bandwidth will likely be insufficient for either safety or non-safety services. To alleviate the spectrum demand, a number of solutions have been proposed. The work in [46] considers vehicles equipped with a DSRC and a UHF radio, and analytically derives the performance gain yielded by a cognitive radio system that allows the use of additional bands. Vehicles equipped with two radios are also considered in [47]. There, Kim et al. introduce a cognitive ad hoc network architecture to allow vehicle opportunistic access to WiFi channels, and present a cognitive routing protocol leveraging geographical location and sensed channel information. A simulation-based study is described in [45], where vehicles sense the UHF spectrum licensed to TV broadcasters and report their measurements to roadside processing units. The latter are in charge of identifying the frequencies available for widening the 802.11p control channel spectrum.

Motivated by the aforementioned observations and studies, in this chapter we focus on the use of low-frequency channels, namely, the UHF band at 700 MHz, in support to the channels at 5 GHz commonly used in ITS. Note that the use of UHF frequencies at 700 MHz for vehicular communications have been already attracting a great deal of interest, initially by the Japanese Transportation Institute and, more recently, by the FCC [48]. Indeed, low-frequency bands offer a significantly larger coverage than 5-GHz DSRC implementations. At an identical transmitter power, a low-frequency signal will have greater range than a high-frequency one, due to decreased free space attenuation and lower absorption by buildings and obstacles. The advantage of using the UHF bandwidth is that control information can be exchanged between vehicles and network infrastructure independently of the coverage provided by roadside radio devices. This translates into the possibility for the vehicles to interact with the ITS in advance, and get ready for the (high-throughput) connectivity with an upcoming roadside device. As a result, the time under coverage of the latter can be fully exploited for data transfers, thus reducing the experienced delay.

We develop a real vehicular testbed, with infrastructure nodes operating in the UHF band as well as roadside units (RSUs) operating at 5 GHz, while vehicles are equipped with both a UHF radio and a 5-GHz radio. An abstract representation of the network scenario in the testbed is depicted in Figure 3.1. We focus on content downloading applications, and design a message protocol that leverages the UHF channel for control information and 5-GHz service channels for data delivery. We then investigate, through our testbed, the benefits of such an approach.

To our knowledge, ours is one of the very few existing vehicular testbeds that exploit white spaces or UHF bands [49]. Furthermore, although in this work we limit our attention to the 700-MHz band and to content downloading, our study could be extended to the case of other low-frequency channels, like those used by Digital Mobile Radio (DMR), as well as to include control messages for the support of safety and other non-safety applications.

3.2 System Scenario

As already hinted, our objective is to devise a fast reservation and scheduling mechanism that can support the transfer of content from a server to moving vehicles, exploiting (i) the longer transmission range of UHF communication to prefetch and schedule the delivery in suitable advance and (ii) the high transmission rates and extensive spacial reuse that communication in ISM bands can afford.

We focus on a roadside network consisting of the following actors, which are supposed to be deployed in an area supporting downloading services for vehicular users.

- *Central Controller (CC)* acting as coordinator between content requests from vehicles and scheduled downloads on the vehicular network.
- *RoadSide Units (RSUs)* providing short-range coverage to send downloaded content to passing vehicles; RSUs are supposed to be connected to the CC either through a wireline or through a wireless multihop connection (hereinafter referred to as "CC-RSU link").
- *Long Range Units (LRUs)* base stations operating on UHF bands, used to collect movement updates and content requests from vehicles.
- *On-Board Units (OBUs)* used by vehicles to request content from the CC through the LRU and to download it from RSUs.

Additionally, we assume that each vehicle has a location device (e.g., a GPS) attached to its OBU and that the CC knows the locations of all RSUs under its control. The appropriate UHF channel is automatically supplied to the OBU by a radio map lookup service available on the OBU itself, possibly integrated with sensing channel information [45,47].

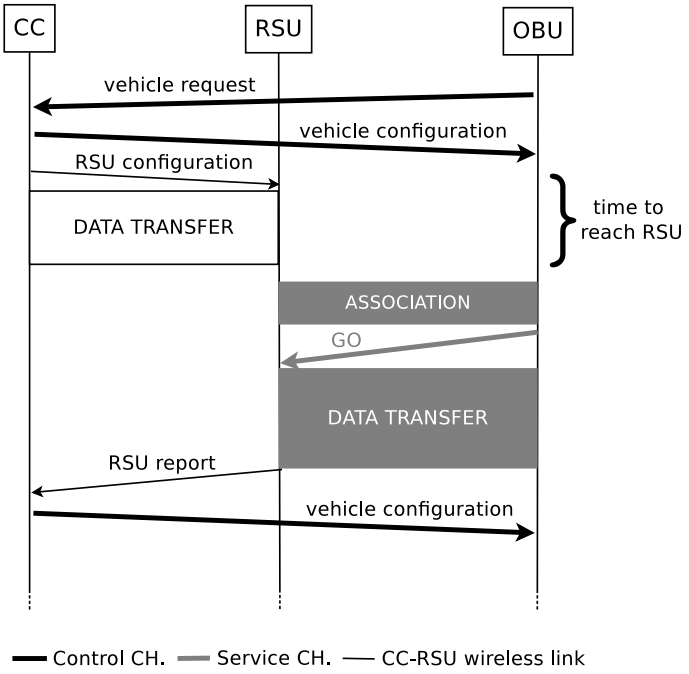


Figure 3.2: Protocol exchange among CC, RSU and OBU.

3.3 Protocol Description

In the following, we describe the protocol interactions between the four actors. We will refer to this protocol as Locate-Fetch-Transfer (LFT), which summarizes the three tenets of its design.

Locate A *Vehicle Beat* message (similar to the CAM specified by ETSI) is broadcast by each OBU every second in the UHF band. This type of message carries geolocation data (latitude, longitude, direction and speed), along with additional (e.g., safety-related) information of a specific vehicle identified by its MAC address. The LRU receives the Vehicle beats and forwards the data to the CC, which then updates each vehicle information and its average speed (computed over the last ten seconds).

Fetch Content is requested by vehicle users through a URL (either provided by the user application, or manually inserted) pointing to an Internet resource or to data locally cached at the CC. The request, along with the MAC address of the requesting vehicle, is received by the LRU on the UHF band and forwarded to the CC. If no LRU is available (the request is not acknowledged), the request is periodically reissued until successful.

When the CC receives the request, it downloads a local copy of the requested content, if not already available. Then, it selects the closest (or the most suitable) RSU to the vehicle from its database and determines if, based on the vehicle position, its predicted

movement, and the expected download rate, the content needs to be split across multiple RSUs along the vehicle path. After identifying the first RSU that the vehicle is likely to come across, the CC sends a *Vehicle Configuration* message toward the vehicle (through the LRU), detailing the network information, such as IP address, netmask, channel and BSSID, needed by the vehicle to connect to the selected RSU. Additionally, the CC partitions the content in one or more macroblocks (depending on the expected number of RSUs involved and on their coverages), and it sends an *RSU Caching* to the RSU nearest to the vehicles. Such a message includes retrieval information for the first macroblock, along with the vehicle ID (e.g., its MAC address). The selected RSU downloads the macroblock through the CC-RSU link, further partitions it into chunks, each of which can fit in a MAC frame, and waits for the vehicle arrival.

Transfer After the OBU of the vehicle has associated to the RSU using the information provided by the Vehicle Configuration message, it starts sending short *Go* messages to the RSU until the first chunk is received from the RSU. The chunks, sent over UDP and with the help of an application-level window protocol, are transferred until either the macroblock is complete, or the vehicle leaves the RSU coverage. When the transfer thus ends, the RSU returns an *RSU report* message to the CC, informing it of the final status of the transfer. The CC can then schedule the next RSU, possibly repartitioning the remaining data of the requested content among one or more macroblocks.

Figure 3.2 summarizes the LFT exchanges upon the issuing of a vehicle request, among three of the four actors: for the sake of simplicity, the communication between CC and OBU is always assumed to go through the LRU.

3.4 Testbed Setup

To validate the framework in a real scenario, we have relied on our TV White Spaces (TVWS) testbed, in the Viù Valley, a mountain area in north-western Piedmont (Italy). There, we have selected a TV frequency that is allocated to a broadcaster, but that is not currently used. We have installed a bidirectional communication system based on the IEEE 802.11 specifications, as described below.

Hardware Configuration

The coverage of the valley is guaranteed through an LRU with the following characteristics:

- central frequency: 763 MHz, channel bandwidth: 5 MHz;
- antenna: 70-degree span, 9 dBi gain;

Figure 3.3: Antenna configuration on the testbed car: 700 MHz (red circle) and 5 GHz (green circle) antennas.

Figure 3.4: Antenna configuration at the RSU: antenna link with CC at the top, and with the OBU at the bottom.

- transmission power: 18 dBm.

Vehicles are equipped with two omnidirectional antennas, at 5 GHz and 700 MHz, respectively; the former has 5 dBi gain and uses a transmit power of 22 dBm, the latter has 6 dBi gain and uses a transmit power of 18 dBm. On the vehicle, we have installed a device with two miniPCI cards, one for the 5-GHz network and the other for the channel at 700 MHz.

RSUs have been installed as APs operating at 5 GHz. As shown in Figure 3.4, RSUs are equipped with two directional antennas (30-degree span, 23 dBi gain) at 5 GHz, one of which is used to handle data exchange towards vehicles, while the other is used for the CC-RSU link.

As described in [50], we compared the performance of the UHF system along the road to that of a device operating at 5 GHz. The wireless cards used the MadWiFi driver, with the Minstrel rate adaptation algorithm activated. We evaluated the received signal strength index (RSSI) and throughput in both bands; the values of RSSI measured at 700 MHz, shown in Figure 3.5, are such that a good data rate is always guaranteed between vehicles and LRU.

Table 3.1: Transfer summary at 20 km/h: worst case, 4 contacts (top), and best case, 3 contacts (bottom)

	RSU1	**RSU2**	**RSU2**	**RSU1**
From CC [chunks]	61728	61728	53053	13163
To OBU [chunks]	57869	57888	39890	13163
Coverage time [s]	71	69	63	26
Throughput [Mb/s]	8.45	8.70	6.56	5.25

	RSU1	**RSU2**	**RSU2**
From CC [chunks]	61728	61728	54334
To OBU [chunks]	54939	59537	54334
Coverage time [s]	85	78	63
Throughput [Mb/s]	6.70	7.91	8.94

Table 3.2: Transfer summary: worst case, 40 km/h

	RSU1 Avg	**RSU2 Avg**	**RSU Global avg**
From CC [chunks]	34681	44456	39025
To OBU [chunks]	16964	20997	18757
Coverage time [s]	36	36	36
Throughput [Mb/s]	4.94	5.96	5.40

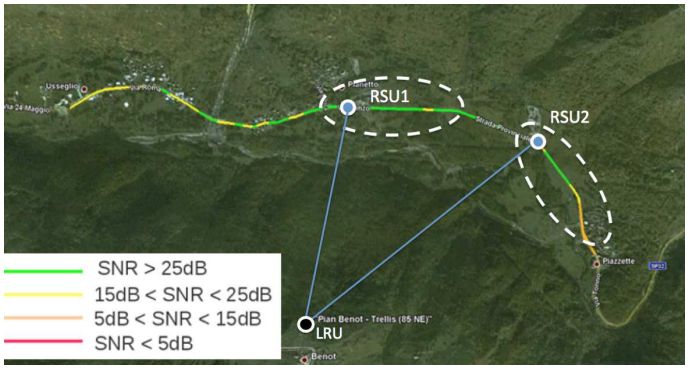


Figure 3.5: RSSI values at 700 MHz in the testbed road.

Table 3.3: Transfer summary: 5 GHz only, worst case, 40 km/h

	RSU1 Avg	**RSU2 Avg**	**RSU Global avg**
From CC [chunks]	31916	33117	32492
To OBU [chunks]	8487	4874	6752
Coverage time [s]	48	39	44
Throughput [Mb/s]	1.85	1.29	1.61

Finally, the OBU aboard the vehicle has two IP addresses. The first one is used for exchange of signaling messages with the LRU and the CC, the second address is dynamically configured, as described above, and is used during the data downloading from the RSUs.

LFT Parameters

We have implemented the LFT protocol described in Section 3.3, and tested it with one vehicle travelling on the stretch of road in Figure 3.5. We installed two RSUs, namely, RSU 1 and RSU 2, operating in the 5 GHz bands using channel 100 and 120, respectively. In order to represent the passage under several RSUs along the road, the vehicle proceeds as follows. It starts outside the coverage of RSU 1, then enters it and associates to the RSU. Next, the vehicle leaves the coverage of RSU 1 and, a little later, enters the coverage of RSU 2 and associates to it. Finally, it leaves RSU 2, turns around and drives back, repeating the procedure in reverse order. The vehicle is driven back and forth until the transfer is complete.

We have run standalone tests where the vehicle uses LFT to request and download a 200-Mbyte file in each experiment. We then compared the attained performance to a case where a 2-Mb/s dummy download was activated from each RSU toward an additional OBU in a parked vehicle.

Each file was split into 168,810 chunks of 1296 byte each. In each test, the vehicle travels either at a steady speed of 20 km/h or 40 km/h. Given the coverage attained with the directional antennas at the RSUs, such vehicle speeds result in the scheduling at RSUs of 61,728 and 45,000 chunks, respectively. If no chunks were lost, 3 and 5 contacts with RSUs, respectively, would have been enough to complete the whole file transfer.

3.5 Experimental Results

We now present and discuss the performance recorded on the previously described testbed. It is worth pointing out that, due to the duration of each test, not many of them could be run in the same environmental conditions (namely, over a few hours' span, meteorological conditions in a mountain valley are bound to change dramatically). Therefore, we could not provide a solid statistical averaging of metrics and we resorted to showing the

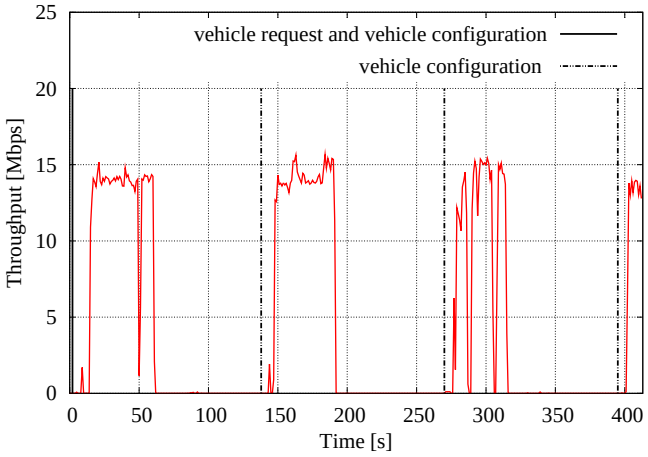


Figure 3.6: Instantaneous throughput vs. time, worst case; speed: 20 km/h.

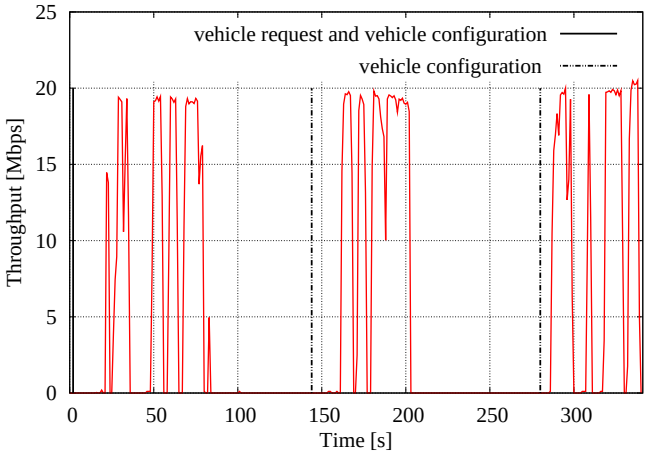


Figure 3.7: Instantaneous throughput vs. time, best case; speed: 20 km/h.

worst-case results recorded across each type of experiment. Occasionally, we will also provide a set of best-case results for the sake of comparison.

The first set of results, showed in Figure 3.6, illustrates the instantaneous application-layer throughput recorded during each of the four contacts with the RSUs (each contact being separated by a vertical line), in the worst recorded case, driving at 20 km/h. Since the signal quality did not allow to exceed 15 Mb/s of throughput, one contact more than necessary had to occur for the entire file to be transferred. It is however to be noted that, in the best case (Figure 3.7), three contacts, as predicted, are enough to complete the transfer, thanks to a sustained throughput of almost 20 Mb/s.

Table 3.1 further details the transfers showing, respectively in each row, the macroblock size scheduled by the CC in anticipation of the upcoming contact[1]; the number of

[1]For clarity, this quantity is expressed in number of chunks; recall, however, that the macroblock is

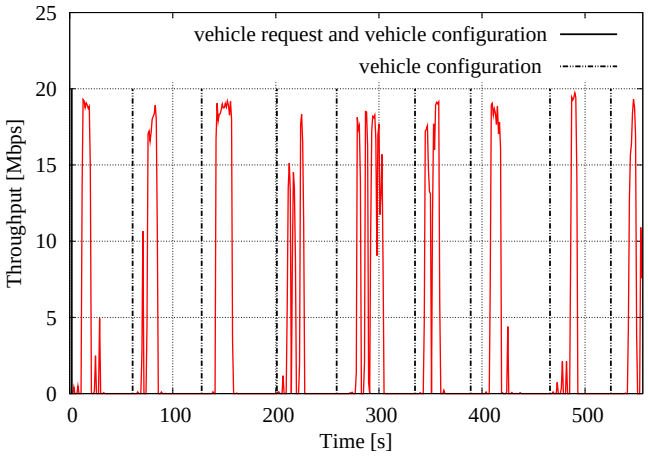


Figure 3.8: Instantaneous throughput vs. time, worst case; speed: 40 km/h.

chunks actually downloaded by the OBU of the passing vehicle; the time under coverage[2] of the RSU and, finally, the average application-layer throughput while under coverage.

In the second set of results in Figure 3.8, the test was repeated while driving at a steady speed of 40 km/h. As expected, coverage under each RSU lasts for a shorter time, hence the greater number of contacts needed. Table 3.2 reports the average number of chunks scheduled and transferred in each of the 9 contacts (though not shown here, 9 contacts were needed in the worst case while 7 contacts in the best case).

The comparison of the previous case with the scenario including background traffic shows a performance degradation, which is mainly due to the additional flow carried on the channel at 5 GHz (results are omitted for brevity).

Finally, we asked ourselves what the impact of the locate-and-fetch components of the LFT protocol is, by comparing it to the case where only 5-GHz bands are used for both control and data messages. Note that, in these tests, the OBU sends the request through the RSU and the downloading of the remaining chunks is negotiated at the time of every contact on the 5 GHz channel. This results in a plain vanilla file transfer lacking the benefits of preemptive feeding of content to the upcoming RSU, as the vehicle has no means to send its updated position to the CC while travelling outside the RSU coverage. As shown by Table 3.3, the average number of chunks transferred to the OBU is reduced, yielding a low throughput. Indeed, precious time under RSU coverage ends up being wasted in negotiating the download of the file leftover. Overall, the average throughput during content downloading resulted to be 5.40 Mb/s, implying a 3x gain yielded by the usage of the UHF band for the transmission of control messages.

divided into chunks only at the RSU.

[2]Here and in the following tables, for the last passage, this value represents the time under coverage till download is complete.

3.6 Conclusion

We defined a protocol for content downloading services, which leverages 5-GHz bands for data delivery and UHF bands for the transmission of control messages (aimed at locating vehicles and collecting requests). We assessed the benefits of exploiting UHF bands, providing much larger coverage than the 5-GHz frequencies, through a vehicular testbed. Our experimental results show that a 3x throughput gain in content delivery can be achieved with respect to the case where only 5-GHz bands are used. Such a gain is due to preemptive data feeding to the upcoming RSU and to the fact that RSU coverage time is fully exploited for high-throughput data transfers.

Future work will expand along the following directions: (i) experimental tests on more complex road topologies, (ii) implementation of a mechanism for dimensioning the content resource to be transferred to the RSU, based on the expected RSSI, (iii) implementation of fast authentication procedures as the vehicle moves in and out the coverage of different RSUs.

Chapter 4

RSUs Deployment in Intelligent Transportation Systems

In this chapter, similarly to the previous one, we study content downloading along with content dissemination, but we focus more our attention on the problem of RSUs deployment to ensure satisfying performance to bypassing users. This topic is quite interesting for the network operators' point of view: in fact, they are strongly motivated in finding the right trade-off between performance and costs. Even if over-dimensioning the network installing a huge number of RSUs could guarantee high-performance, this is not the best solution for the operators that constantly try to limit the installation and management costs of the infrastructure.

Thus, in this chapter we envision new RSUs deployment strategies able to find the right trade-off between costs and performance. The problem is solved through mathematical modeling and optimization; then, the efficacy of the proposed strategies have been evaluated using simulations with realistic settings. Intermittent connectivity of the urban area is provided, although good service quality for both dissemination and content downloading is guaranteed.

The content of this chapter is organized as follows. In Section 4.1 we illustrate the RSUs deployment problem. In Section 4.2 we review existing approaches to the problem of optimal RSUs deployment, and we discuss them by highlighting the differences and the performance they can achieve. In Section 4.3 we present different formulations of the infrastructure deployment problem. In Section 4.4 we describe the real-world mobility scenarios we use in order to study the performance of content dissemination and downloading. In Section 4.5 we use the aforementioned heuristics to derive an optimal RSUs deployment in the realistic mobility scenarios under study. In this case, ideal channel access and propagation conditions are considered and no protocol aspects are accounted for. Simulation results obtained through the network simulator ns2 are presented in Section 4.6. Again, real-world road topologies and vehicular traces are used. The optimal RSUs

deployment, computed as outlined above, is taken as an input to the simulation, and vehicular users are assumed to either receive the same information (dissemination service) or download a different information item each, from the fixed Internet through 802.11p RSUs. Finally, in Section 4.7 the major lessons learnt through the presented study are summarized and some guidelines for RSU deployment in ITSs are highlighted.

4.1 Problem Statement

Most carmakers are striving to create an in-vehicle environment which is as comfortable and entertaining as possible. An high percentage of newly-manufactured vehicles boast multimedia capabilities that were once thought to belong to a living room, like LCD screens or gaming consoles. Such technological wealth, however, is not complemented with live features besides radio broadcasts. The presence of multiple LCD screens for passengers begs, as it were, for advanced infotainment services of various nature, ranging from email/social network access to more bandwidth-demanding contents, such as newscasts or local touristic clips. Without affecting drivers attention, navigational aids may be integrated by short videos showing traffic congestion and recommending alternate routes. Furthermore, in keeping with the explosive growth of social networks, it is envisioned that car passengers may show a high interest in car-oriented social networking and multiplayer games. Finally, professional drivers could access services for efficient vehicle fleet coordination, up-to-the-minute updated goods deliveries or re-routing, and customized cab pick-ups. Currently, the only connectivity option for vehicles amounts to accessing a 3G network, which could provide high-speed network availability but is hampered by restricted competition among network operators. Also, the lack of a local infrastructure, which is specifically dedicated to geolocalized services, makes the realization of the above scenarios hard to implement and limits its features. However, the emergence of communication standards for vehicular networks is bringing new visions and opportunities that could come close to the always-connected paradigm. Globally referred to as Intelligent Transportation System (ITS), this new vision aims at improving transportation in terms of safety, mobility, traffic efficiency, impact on the environment, and productivity. Motivated by such a vision, this chapter deals with the dissemination of information from RSUs to vehicular users within a geographical region, as well as the downloading from RSUs of delay-tolerant (e.g., map services, touristic information) and bandwidth-demanding (e.g., video streaming) content, by passing-by vehicles. More specifically, the presented study tackles the issue of deploying an ITS infrastructure based on the IEEE 802.11p technology, which efficiently achieves the goal of information dissemination and downloading in spite of the fleeting connectivity, highly dynamic traffic patterns, and constrained node movements. To this end, the following key issues are investigated:

i) Assuming that an area, with an arbitrary road topology, must be equipped with a limited

number k of infrastructure nodes, what is the best deployment strategy to maximize the dissemination process or the downloading throughput?

ii) Given such an optimal RSU deployment, what is the actual throughput performance that can be achieved by users when realistic traces are used to represent the vehicular mobility?

The answers to the above questions are given in the remainder of the chapter.

4.2 Related Work

Several works in the literature have addressed the problem of the deployment of RSUs for vehicular access, although with a number of significantly different objectives. In this chapter, the focus is on RSU deployments for (i) the dissemination of information to all vehicles in a geographic region, and (ii) the downloading of content from Internet-based servers, by a subset of the vehicles.

The simplest solution possible to the RSU placement problem, i.e., a random deployment, is evaluated in [51]: such a strategy, representative of unplanned access networks like those identified in the real world [52], is shown to help the routing of data within urban vehicular ad-hoc networks. Similarly, intuitive RSU deployments, that are not justified by means of a theoretical or experimental analysis, are evaluated in [55], with the goal of improving delay-tolerant routing among vehicles, by letting each AP work as a static cache for contents that have to be transferred between vehicles visiting it at different times. However, although they benefit from the routing process, random or intuitive placements cannot represent, in general, an optimal solution to the RSU deployment problem. In [53], [54] and [60], it is demonstrated that such simple strategies are easily outperformed by more sound approaches, for both the dissemination and the downloading objectives. As a consequence, in the following the focus will be on deployment strategies that are instead built upon a precise placement rationale.

Firstly, note that standard maximum graph coverage approaches, such as those adopted in [65], [66], [67], [68] and [69] do not fit the RSU deployment problem as considered in this chapter. Indeed, these placement strategies are designed for sensor or cellular networks, and thus assume that the infrastructure nodes form a connected network or provide a continuous coverage of the road topology. Moreover, many standard infrastructure deployment techniques have an energy efficiency goal that instead is not of interest in a vehicular environment. Secondly, the mobility of vehicles is rather unique, as it obeys traffic regulations, is constrained by the street layout and alternates very high and very low speeds in relatively short times pans. As a consequence, the scenario differs significantly from those studied in [70], which deals with the deployment of Internet access points in a static network, and in [71], which targets a mobile sensor network.

In [58] and [59] target the maximization of the amount of time each vehicle is within range of at least one RSU, an objective that may be seen as beneficial for both the dissemination and downloading objectives. More precisely, these works formulate optimization problems whose solutions provide the RSU positions that maximize the coverage time. In the work [58], a minimum coverage requirement is guaranteed, while that by [59] maximizes the minimum contact opportunity. However, both RSU deployment strategies assume that a predefined set of paths over a given road topology is provided, which makes their application limited to the particular case where only a subset of the total traffic is to be covered by RSUs.

Within the context of information dissemination to all vehicles in a geographical region, [57] recently formulated an optimization problem that aims at maximizing the spreading of an information within a temporal horizon. However, this work assumes that RSU positions are given, and thus does not address the issue of the identification of the RSUs locations. Recently, multiple dissemination schemes have been evaluated in the context of vehicular environments in [64], with the aim of leveraging opportunistic vehicle-to-vehicle communication so to offload the cellular infrastructure from the need of forwarding some small information to all vehicles in a geographical region. However, again, this work does not cope with the placement of RSUs, since the access network is represented by a pervasive and ready-to-use cellular network. In [56], the authors formulate an optimization problem for the planning of RSU locations, solvable with genetic algorithms. However, the deployment is intended to facilitate the aggregation of data, collected by the vehicles, on the road traffic conditions, rather than the dissemination of information. [54] describes an optimization problems for the deployment of RSUs, whose objective is that of the dissemination of information to vehicles in the shortest time possible. In this chapter, the formulation and results of this work are discussed and employed as the starting point for a simulative performance evaluation campaign.

RSU deployments that aim at maximizing content downloading are instead proposed in [53] and [60]. However, the associated optimization problems are designed for a cooperative downloading, i.e., a process where direct RSU-to-vehicle data exchanged are augmented through vehicle-to-vehicle communication: the cooperative downloading thus leverages opportunistic contacts among vehicles to increase the downloading speed. Of the two, the formulation in [60] is the most complete, measuring the actual per-user throughput in presence of realistic data transmission rates, channel access and interference. Also, [61] adopts a theoretical framework to study the RSU deployment density that minimizes the uploading delay via vehicle-to-vehicle multi-hop communication. However, RSU placement strategies that assume cooperative approaches and vehicle-to-vehicle data transfers will not be further discussed in this chapter, since the focus here is on the downloading via direct RSU-to-vehicle communication.

Also, related to the transfer of data in vehicular environments are the work [62] and [63]. The former deals with the collection at the RSUs of small-sized data generated

by vehicles, and thus targets content uploading rather than the dissemination and downloading, which are addressed in this chapter. The latter focuses instead on the scheduling of data packets at RSUs, while it assumes the infrastructure deployment to be given: therefore, it does not concern the RSU placement problem considered here.

4.3 RSU Deployment for Content Dissemination and Downloading

In this section, the problem of planning vehicular networks for information dissemination and downloading is studied taking into account the peculiarities of the vehicular environment. In order to capture both the dissemination and downloading applications with a single framework, the problem is cast as that of deploying a limited number of RSUs so as to maximize (i) the number of vehicles served (i.e., covered) by the RSUs, and (ii) the connection time between vehicles and RSUs. Such an approach fits well both the targeted applications, which can be modeled as separate instances of the same problem above, characterized by different durations of the desired connection time. As a matter of fact, on the one hand, a dissemination process typically concerns small pieces of information and large amounts of vehicles, thus it requires that as many vehicles as possible enjoy a small connection time. On the other hand, the downloading application limits the number of mobile users involved in the process, since only a fraction of them is interested in retrieving some content from the Internet at the same time; however, each of such vehicles must be covered for a long time, so as to be able to download the whole amount of the data it demanded.

RSU Deployment as an Optimization Problem

Formally, an urban road topology of area size equal to A is considered, including N intersections. RSUs can be deployed at any of the N intersections, since, as proved in [54], placing RSUs at road junctions yields significant advantage over positioning them along road segments, in terms of both covered vehicles and connection time. Each RSU is assumed to have a service range equal to R. Such a service range may map into the transmission range of the RSU, or into a multiple of its transmission range if information delivery can be performed through multi-hop communication. Also, denote by V the number of vehicles that transit over the area A during a given time period, hereinafter called observation period. The goal is then to deploy k RSUs so as to maximize the number of covered vehicles, among the possible V, so that the connection time between vehicles and RSUs is above a given threshold τ. Notice that this problem can be seen as a generalization of the well-known Maximum Coverage Problem (MCP), as also detailed next. However, the fact that vehicles may be covered by more than one RSU during their

route to destination, jointly with the connection time constraint embodied by the threshold τ, makes most of the generalizations to the MCP unsuitable to the problem studied in this chapter. Therefore, in the following several solutions to the problem outlined above are discussed, as presented in [54].

Contact-Only RSU Deployment

As a baseline strategy to compare with, an RSU deployment is first introduced that only considers the number of vehicle-to-RSU contacts, while it neglects the connection time aspect. The goal is then to maximize the number of vehicles covered by k RSUs, that can be deployed at the N intersections located in the road topology. To that end, by analyzing the vehicular mobility in the selected area, define an $N \times V$ matrix P whose generic element is given by

$$P_{ij} = \begin{cases} 1 & if\ the\ vehicle\ i\ crosses\ the\ junction\ j\ during\ the\ observation\ period \\ 0 & otherwise \end{cases}$$

The problem is modeled as a Maximum Coverage Problem (MCP), which can be formulated as follows. Given a collection of sets $S = \{S_1, S_2, \ldots, S_N\}$, where each set S_i is a subset of a given ground set $X = \{X_1, X_2, \ldots, X_V\}$, the goal is to pick k sets from S so as to maximize the cardinality of their union. To better understand the correspondence with the problem posed above, consider that the elements in X are the vehicles that transit over the considered road topology during the observation period. Also, for $i = 1, 2, \ldots, N$, we have

$$S_i = \{x_j \in X,\ j = 1, \ldots, V\ :\ P_{ij} = 1\} \tag{4.1}$$

i.e., S_i includes all vehicles that cross intersection i at least once over the observation period. Thus, by solving the above problem, the set of k intersections where an RSU should be placed can be obtained so as to maximize the number of covered vehicles. Unfortunately, the MCP problem is NP-hard; however, as reported in [72], it is well known that the greedy heuristic achieves an approximation factor of $1 - (1 - 1/m)^m$, where m is the maximum cardinality of the sets in the optimization domain. The greedy heuristic, hereinafter also called MCP-g, picks at each step a set (i.e., an intersection) maximizing the weight of the uncovered elements. Now, consider an auxiliary collection of sets G, subset of S, and let W_i (with $i = 1, \ldots, N$) be the number of elements covered by S_i, but not covered by any set in G. The steps of the greedy heuristic are reported in Figure 4.1.

We stress that, although the MCP-g algorithm provides a very good approximation of the optimal solution, it requires global knowledge of the road topology and network system, as well as the identity of the vehicles that have crossed the N intersections during the observation period.

The MCP-g heuristic

Require: k, $\mathbf{P}$, $\mathscr{S}$
1: $G \leftarrow \emptyset,\ C \leftarrow 0,\ U \leftarrow \mathscr{S}$
2: $W_i = \sum_{j=1}^{V} \mathbf{P}_{ij},\ i = 1, \ldots, N$
3: **repeat**
4: Select $S_i \in U$ that maximizes W_i
5: $G \leftarrow G \cup S_i$
6: $C \leftarrow C + 1$
7: $U \leftarrow U \setminus S_i$
8: $W_i = \sum_{\substack{j=1 \\ j : x_j \notin G}}^{V} \mathbf{P}_{ij}, i = 1, \ldots, N$
9: **until** $C = k$ or $U = \emptyset$

Figure 4.1: Algorithm for the greedy MCP heuristic.

Coverage Time-Based RSU Deployment

Next, consider the case of actual interest, i.e., k RSUs have to be deployed at the road intersections so as to favor both the number of covered vehicles, as well as the time for which they are covered. To this end, define an $N \times V$ matrix T whose generic element, T_{ij}, represents the total time that vehicle j would spend under the coverage of an RSU if the RSU were located at intersection i, i.e., the contact time between a vehicle j and an RSU located at intersection i. Then, the following problem can be formulated, named Maximum Coverage with Time Threshold Problem, or MCTTP: given k RSUs to be deployed, the aim is to serve as many vehicles as possible, for (possibly) at least τ seconds each, i.e.,

$$\begin{aligned} &\max \sum_{j=1}^{V} \min\Big\{\tau,\ \sum_{i=1}^{N} T_{ij} y_i\Big\} \\ &s.t.\ \sum_{i=1}^{N} y_i \leq k;\ y_i \in \{0,1\}\ \forall i \end{aligned} \tag{4.2}$$

Note that in the first equation above an RSU is placed at an intersection so as to maximize the number of vehicles that are covered, taking into account a vehicle contact time up to a maximum value equal to τ. RSUs that provide coverage for at least τ seconds to a given vehicle do not further contribute to the overall gain of covering such a vehicle. The constraint in the second equation instead limits the number of RSUs to k. It can be easily verified that the MCP is a particular case of the above formulation, obtained by setting $\tau = 1$ and $T_{ij} = P_{ij}$. Hence, MCTTP is NP-hard and a greedy heuristic is proposed for its solution, denoted by MCTTP-g. The heuristic picks an intersection at each step so as to maximize the provided coverage time, although only the contribution due to vehicles for which the threshold τ has not been reached is considered. Let G, subset of S, be a collection of sets and let now W_i (with $i = 1, \ldots, N$) be the total contact

time provided by intersection i, considering for each vehicle a contribution such that the vehicles coverage time due to the union of the G and S_i sets does not exceed the threshold τ. The greedy heuristic is reported in 4.2.

The MCTTP-g heuristic

Require: k,$\mathbf{T}$,τ,$\mathscr{S}$
1: $G \leftarrow \emptyset, C \leftarrow 0, U \leftarrow \mathscr{S}$
2: $t_j = 0, j = 1, \ldots, V$
3: **repeat**
4: $W_i = \sum_{j=1}^{V} \min(\tau - t_j, \mathbf{T}_{ij}), i = 1, \ldots, N$
5: Select $S_i \in U$ that maximizes W_i
6: $G \leftarrow G \cup S_i$
7: $C \leftarrow C + 1$
8: $U \leftarrow U \setminus S_i$
9: $t_j = \min(\tau, t_j + \mathbf{T}_{ij}), j = 1, \ldots, V$
10: **until** $C = k$ or $U = \emptyset$

Figure 4.2: Algorithm for the greedy MCTTP heuristic.

Again, notice that the time-threshold heuristic requires knowledge of the global road topology and of the vehicles identity. While the first assumption appears realistic, the second may not always be so: therefore, it needs to be relaxed, proposing an RSU deployment strategy that is unaware of the vehicles identities.

Absence of Information of Vehicles Identities

When the vehicles identities are not available, the only information that can be exploited is the total time that all vehicles would spend under the coverage of an RSU if it were located at intersection i, i.e.,

$$T_i = \sum_{j=1}^{V} T_{ij} \quad i = 1, \ldots, N \tag{4.3}$$

Thus, in this case the objective is to maximize the total contact (service) time offered to vehicles when k RSUs are deployed. In this case, the problem can be formulated as a 01 Knapsack Problem (KP), which is defined in [73] as follows. A set of N intersections (items) $I = \{I_1, I_2, \ldots, I_N\}$ is given; each intersection has a value T_i and unitary weight, and the maximum number of selected intersections (maximum weight) must be equal to k. The goal is to select a subset of k intersections that maximizes the overall service time provided to vehicles, i.e.,

$$\begin{aligned} &\max \sum_{i=1}^{N} T_i y_i \\ s.t. \ &\sum_{i=1}^{N} \leq k; \ y_i \in \{0,1\} \ \forall i \end{aligned} \tag{4.4}$$

The 01 KP is an NP-hard problem in general; however in the case under study, where each intersection has a constant weight, it can be solved in polynomial time by simply sorting the intersections in decreasing order by their value, and selecting the first k intersections. This algorithm, which requires the knowledge of the T_i coefficients (with $i = 1, \ldots, N$), is referred to as KP-T.

Computational Complexity

The computational complexity of both MCP and MCTTP is $O(VN^k)$: given N intersections, all possible combinations where the k RSUs can be placed have to be considered and the weight of each intersection is computed over V vehicles. The cost of both greedy heuristics, MCP-g and MCTTP-g, is $O(KVN)$, since, for k times, the best choice among the candidate intersections (initially set to N) has to be selected, and again the selection is based on the weight computed over V elements. The complexity of the algorithm to solve the 01 KP is $O(VN + N \log N)$, since it is enough to consider each of the N intersections and sort the values to obtain the best k choices.

4.4 Mobility Scenarios

In order to carry out the performance evaluation of the information dissemination and downloading services, real-world road topologies from the canton of Zurich, in Switzerland, are used. Realistic traces of the vehicular mobility in such a region, generated by the Simulation and Modeling Group at ETH Zurich, are made available in [74]. These traces describe the individual movement of cars through a queue-based model calibrated on real data: as detailed in [75], they provide a realistic representation of vehicular mobility at both microscopic and macroscopic levels.

The four road topologies depicted in 4.3(a)-4.3(d) are considered; they represent $100\ km^2$ portions of the downtown urban areas of the cities of Zurich, Winterthur, and of the suburban areas of Baden and Baar. For each topology, half an hour of vehicular mobility is extracted, in presence of average traffic density conditions.

In order to remove partial trips (i.e., vehicular movements starting or ending close to the border of the square area), the trace is filtered by removing cars that traverse only three intersections or less, as well as those spending less than 1 minute in the considered region.

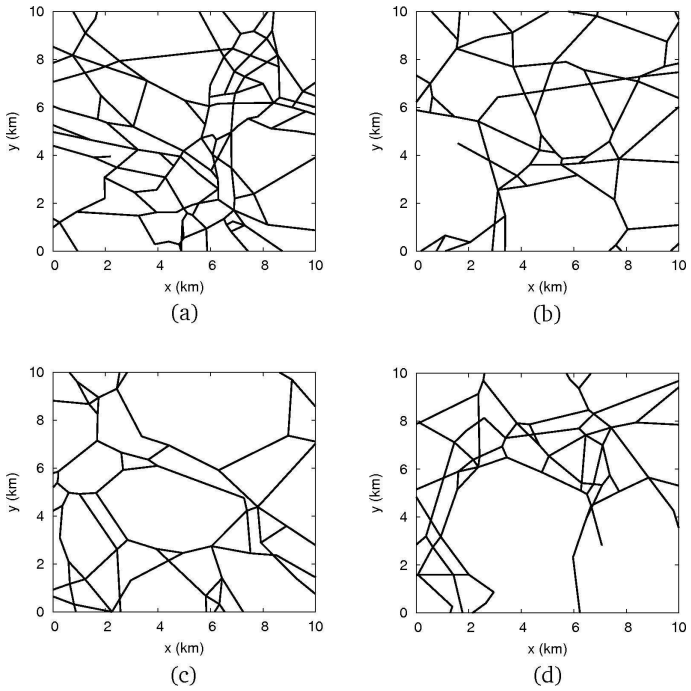


Figure 4.3: Road topology layout: Zurich (a), Winterthur (b), Baden (c), Baar (d).

The selected thresholds result in a low percentage of cars being removed from the traces of the scenarios characterized by a higher traffic density (Zurich and Winterthur), while the filtering is heavier on the traces of the suburban scenarios (Baden and Baar), where the conditions set above are harder to meet. However, the resulting numbers still guarantee the statistical validity of the tests conducted over all road topologies. Specifically, we have 83, 43, 38 and 46 intersections and a number of vehicles equal to 21373, 4942, 5914 and 3736, in the Zurich, Winterthur, Baden and Baar scenarios, respectively.

4.5 Performance Analysis of the Heuristic under Ideal Network Settings

In this section, a first statement on the performance of the algorithms previously introduced is provided. To this end, the different heuristics are solved in presence of real-world road topologies, assuming ideal conditions from a network engineering viewpoint, i.e., no channel losses, ideal disc-like propagation with radius equal to 100 m, perfect medium access, and instantaneous vehicle-to-RSU communication without any need for control messages. The resulting RSU deployments are evaluated in terms of information dissemination capabilities. In the following, the results obtained with the different deployment algorithms maximizing coverage and contact times are compared.

Heuristic Performance

The RSU placement provided by the coverage time-based heuristics presented earlier varies depending on the threshold τ. Indeed, this parameter allows the control of the objective of the deployment, so as to favor

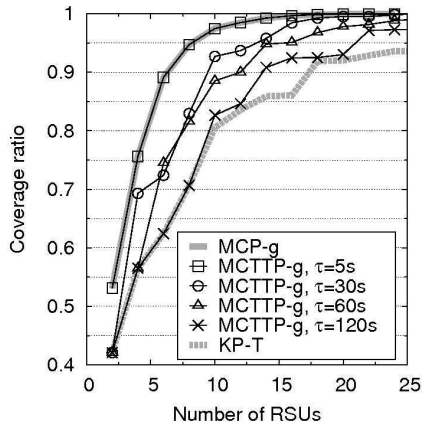
- a dissemination application, by choosing a low value of τ and thus maximizing the number of short-lived contacts that are needed to spread a small content from the RSUs to as many passing-by vehicles as possible;
- a downloading application, by imposing a high value of τ and thus maximizing the number of vehicles that are covered for the arbitrarily long time periods needed to retrieve large-sized files from Internet-based servers.

As a result, let us first observe how the performance of the quasi-optimal deployment obtained through the MCTTP-g algorithm varies as a function of the τ parameter. Figure 4.4(a) focuses on the Zurich road topology and reports the coverage ratio, i.e., the fraction of vehicles that are covered by at least one RSU during their route throughout the scenario, versus the number k of deployed RSUs. The parameter τ ranges between 5 and 120 s, and the plot shows the result of the MCTTP-g scheme along with those obtained under the MCP-g and KP-T solutions. Looking at the results, it can be seen how MCTTP-g falls in between an algorithm that maximizes vehicle-to-RSU contacts, i.e., MCP-g, and one that maximizes the overall coverage time, i.e., KP-T. In particular, for low values of τ, MCTTP-g tends to MCP-g, since the time constraint is easily satisfied (a contact with a single RSU is often sufficient to reach the desired coverage time) and the algorithm can thus focus on maximizing the coverage. Instead, when τ is high enough, MCTTP-g tends to KP-T, since the desired coverage time is seldom reached, and thus the same vehicles end up contributing to the optimization: the focus of the algorithm then shifts onto coverage times.

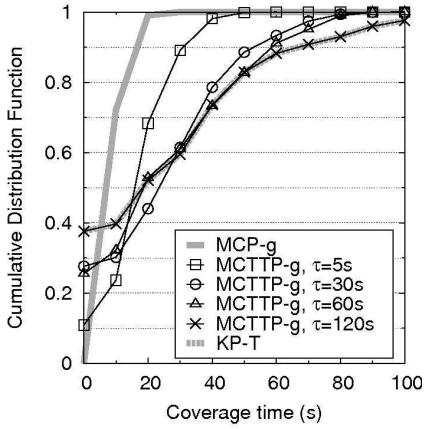
A confirmation to this analysis comes from the CDFs of the per-vehicle coverage time, in Figure 4.4(b), where the same behavior of the MCTTP-g algorithm is observed, as τ varies and for $k = 6$. It can be noted, however, how MCTTP-g with $\tau = 5$ s matches MCP-g in terms of coverage ratio, but outperforms it in terms of coverage time. Similarly, MCTTP-g with $\tau = 120$ s matches KP-T as far as the coverage time is concerned, but provides a better coverage ratio. The combined maximization of contacts and coverage time can thus achieve better performance than contacts-only, or time-only driven solutions, even in borderline conditions.

Thus, it can be concluded that the coverage time threshold τ can be leveraged to calibrate the RSU deployment so as to fit the goals of the different types of services.

Next, it is important to evaluate the role that different mobility scenarios play in the RSU deployment problem. This aspect is evaluated by fixing the threshold τ to 30 s, a contact duration that should be largely sufficient to transfer a few kbytes from RSUs to



(a)



(b)

Figure 4.4: Coverage ratio versus number deployed RSUs (a) and CDF of the coverage time for 6 deployed RSUs (b), in the Zurich scenario.

vehicles. The MCTTP-g algorithm is compared to the optimal solution to the original MCTTP formulation, which, as mentioned, is NP-hard and thus solvable only for small instances of the problem (in the case under study, up to $k = 6$) via a brute-force approach. The outcome of the KP-T algorithm is reported as well, along with that of a random RSU deployment, so to benchmark the performance of the other schemes. The coverage ratio achieved by such algorithms, in different road topologies and as the number of deployed RSUs varies, is depicted in the plots of Figures 4.5(a)-4.5(d). Observe that, regardless of the road topology considered, the MCTTP-g solution is always extremely close to the optimal one. Moreover, the availability of information on the vehicular mobility plays a major role in favoring contacts among vehicles and RSUs: as a matter of fact, the random solution performs poorly, the KP-T algorithm provides an improved coverage of the

vehicles, but the best performance is achieved by the MCTTP-g scheme, which leverages the most detailed knowledge of the vehicular trajectories. Such a result is consistent throughout all scenarios, although the entity of the difference in the coverage ratio provided by the different deployment algorithms varies with the considered road topology. More precisely, a more complex road topology, such as that of the Zurich area, leads to more significant differences between the schemes that are mobility-aware and those that are not.

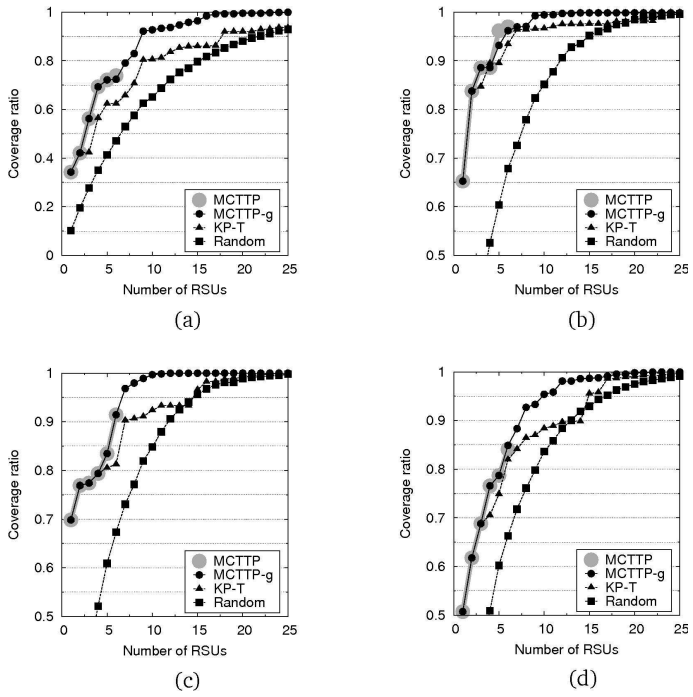


Figure 4.5: Coverage ratio versus the number deployed RSUs, in the Zurich (a), Winterthur (b), Baden (c) and Baar (d) scenario.

Figures 4.6(a)-4.6(d) report instead the distribution of the coverage time, in the specific case in which $k = 6$ RSUs are deployed over each road topology. Recall that the goal is to maximize the time spent by vehicles under coverage of RSUs, up to the threshold time τ of $30\ s$, identified by the vertical line in the plots. The common result in all road topologies is that random deployments lead to small coverage times, whereas the other schemes tend to behave similarly, although KP-T is characterized by a more skewed distribution than those of the MCTTP, in both its optimal solution and greedy approximation. As a matter of fact, the deployments determined by KP-T result, at a time, in more vehicles with very low coverage times, and more vehicles with very high coverage times. Conversely, the MCTTP leads to more balanced distributions, where many vehicles experience a coverage time around the threshold τ. Once more, these observations hold for all the scenarios considered.

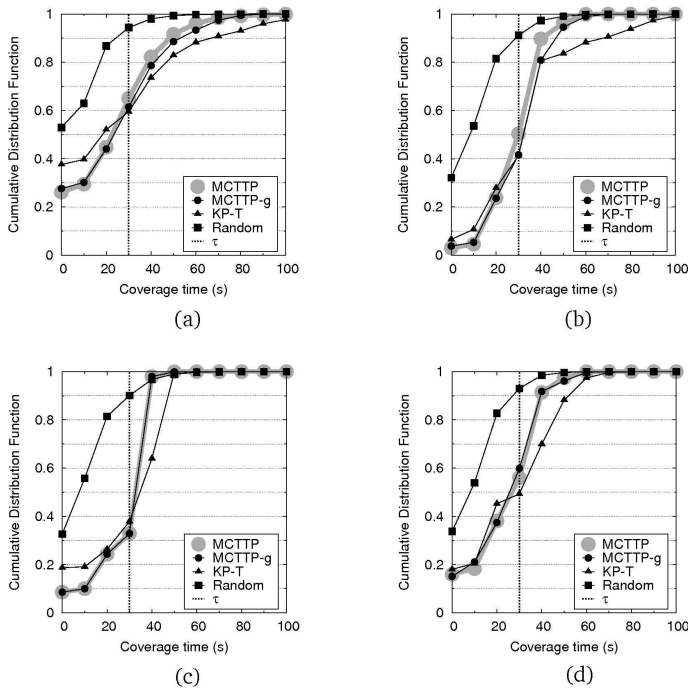


Figure 4.6: CDF of the coverage time for 6 deployed RSUs, in the Zurich (a), Winterthur (b), Baden (c) and Baar (d) scenario.

When comparing the coverage times in Figures 4.6(a)-4.6(d) with the corresponding coverage ratios in Figures 4.5(a)-4.5(d), notice that MCTTP and MCTTP-g provide very similar performance, generally superior to those achieved by the other schemes. Indeed, a random deployment of RSUs induces both a lower number of vehicle-to-RSU contacts and a shorter coverage time with respect to the solutions above. The KP-T solution leads to a performance comparable to that of MCTTP and relative heuristics in terms of coverage time, although with the skewness discussed before; however, this result is paid at a high coverage ratio cost.

In conclusion, the MCTTP formulation represents an efficient solution to the RSU deployment problem which enjoys the desirable properties of (i) being configurable to a specific application by properly setting the value of the parameter, (ii) having an inexpensive greedy heuristics that well approximates the optimal solution, and (iii) yielding results that are consistently better than those achieved with unplanned RSU placements or Knapsack Problem-based formulations.

Impact of Routes

The figures in the previous section are averaged over space, aggregating the coverage ratio and time of all vehicles moving in the region under study. As a further step in the analysis, observe how the route traveled by a car affects the coverage it enjoys during its

movement. To that end, the MCTTP heuristics is considered and the coverage time on a per-route basis is computed by aggregating the performance referring to vehicles that followed the same route through the road topology scenario.

Figures 4.7(a)-4.7(d) show the coverage time measured on different routes traveled by vehicles, in the different mobility scenarios, when the threshold τ is set to $30\ s$ and the number of deployed RSUs is varied. The x-axis of the plots reports the route identifier, which is assigned according to a decreasing coverage ratio ordering (which implicitly leads to monotonically decreasing curves). Observe that when just one RSU is placed in a region, the coverage time is around $20\ s$ (i.e., approximately the duration of one vehicle-to-RSU contact) along a subset of routes (i.e., those passing by the location of the lone RSU). This result is consistent for all the road scenarios, although the number of routes with non-zero coverage time varies depending on the street layout: clearly, more complex topologies imply that more combinations of consecutive segments are available, and thus that a higher number of possible routes will pass by the deployed RSUs. When the number of deployed RSUs increases, as one could expect, more routes become covered for a longer time. However, note that disparity among routes grows along with the number of RSUs deployed: when more RSUs become available, the luckiest routes tend to get coverage durations that are 5 to 10 times those experienced by vehicles traveling on the less fortunate routes. This result evidences how some routes are more prone to enjoy better coverage than others, even in presence of a coverage that is approximating the optimal one.

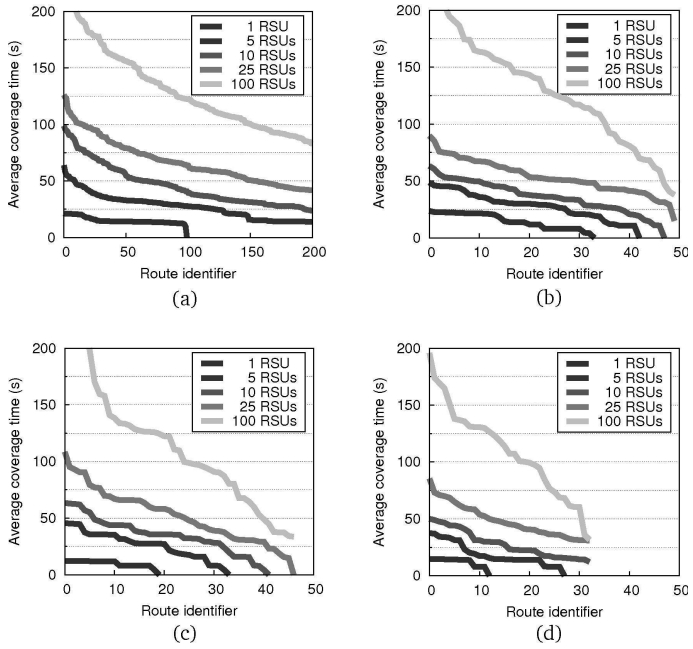


Figure 4.7: Average coverage time versus route, for a varying number of RSUs, in the Zurich (a), Winterthur (b), Baden (c) and Baar (d) scenario.

The impact of the time threshold τ on the per-route performance is instead evaluated in Figures 4.8(a)-4.8(d), when the number of RSUs is fixed to 10. It is quite evident that lower values of τ allow a fairer distribution of RSUs over the road topology, as more routes are covered, even if for a shorter amount of time on average. On the contrary, increasing the τ threshold forces an RSU deployment that is significantly more clustered, with the result that a smaller subset of routes enjoys a high coverage time, while the rest is left uncovered or almost so. This behavior can be observed over all road topologies, but it is especially evident in the Zurich scenario, due to the larger choice of routes enabled by the more complex street layout. A visually-intuitive representation of such a disparity is provided in Figures 4.9(a)-4.9(d): the plots show maps of the four road scenarios, where darker and thicker lines represent road segments traveled by vehicles that have higher coverage times. The results refer to the case of 10 RSUs and τ equal to 30 s, but similar figures were obtained under any other parameter combination. According to these results, it can be concluded that RSU deployments can be significantly unfair, and, as a consequence, that a given average coverage time does not necessarily mean that all vehicles will spend such a time interval under coverage of RSUs. Indeed, especially when the number of RSUs or the minimum time constraint grow, a dramatic disparity can emerge in the performance observed by individual vehicles traveling in the same region but along different routes.

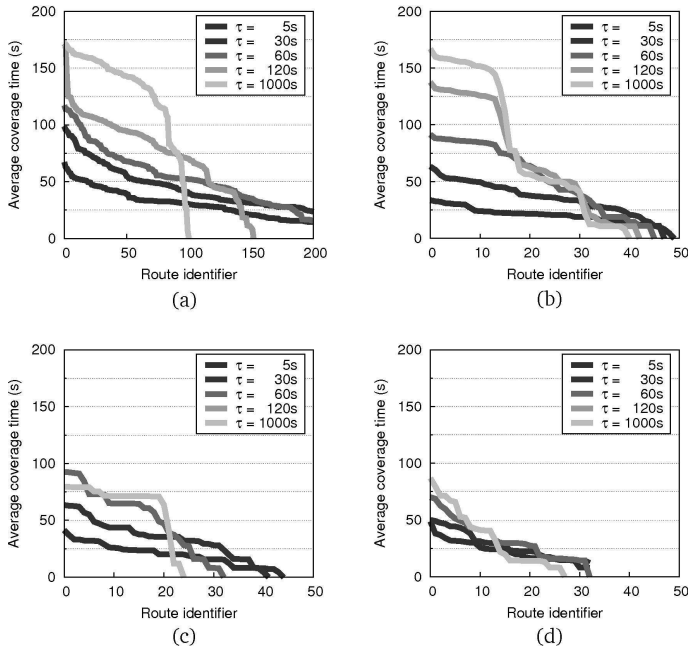


Figure 4.8: Average coverage time versus route, for a varying threshold τ, in the Zurich (a), Winterthur (b), Baden (c) and Baar (d) scenario.

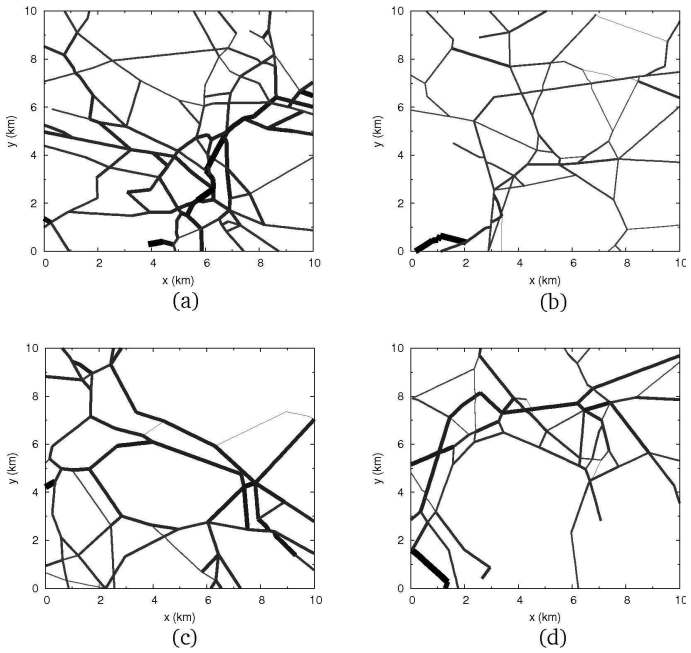


Figure 4.9: Per-road coverage time, for 10 RSUs and $\tau = 30\ s$, in the Zurich (a), Winterthur (b), Baden (c) and Baar (d) scenario.

4.6 Performance Analysis of Heuristic in Realistic Environments

In order to provide a realistic assessment of our heuristics, we ran ns2 simulations tracking vehicle movements in the four maps. Each vehicle is assumed to be equipped with an IEEE 802.11p interface with which it communicates with RSUs. All RSUs use the same frequency channel, 20-MHz wide, for beacons (issued every $0.2\ s$) and any other communication with the vehicles. For simplicity, the link between vehicles and RSUs is established on the service channel, operating at a data rate of $54\ Mb/s$, which, due to the channel switching of IEEE 802.11p, translates to an effective maximum throughput of about $13\ Mb/s$ at the application layer, in absence of contention and of transmission errors. The link is simulated according to a shadowing model defined in [76], with urban parameters for the Zurich and Winterthur maps (resulting in shorter-range coverage), and with suburban parameters for the Baded and Baar maps (resulting in a longer-range coverage). For each scenario, the transmission power of vehicles and RSUs is set in such a way that 95% of the transmitted packets are correctly received at a distance of $100\ m$.

In the dissemination case, the information is included with the beacon issued by the RSU (the beacon size is increased from 22 to 1000 bytes), while in the downloading case the vehicle sends a request packet upon receiving the first beacon from an RSU. The request packet specifies the size of the file to be downloaded. Each downloader wishes

to retrieve a file whose size follows the experimental distribution derived in [77]. Each file is divided into chunks of 1400 bytes each and the RSU starts sending it as soon as the request is received. Chunks carry application-layer sequence numbers, thus vehicles can selectively request the retransmission of missing chunks.

We start by looking at the dissemination case. Based on the discussion presented in the previous section, the parameter τ is set to $30\ s$, and RSUs are placed on each map according to the MCTTP-g heuristic. Several experiments are carried out for each of the four road layouts, with a number of RSUs ranging from 5 to 25. Figure 4.10(a) shows the coverage ratio, computed as the ratio between the number of vehicles that receive at least one beacon and the total number of vehicles in the area. Note that the results match the behavior of the coverage ratio metric obtained in the various scenarios under ideal settings (see Figures 4.5(a)-4.5(d)), although lower performance is achieved due to the fact that realistic propagation conditions are now modeled. Interestingly, though being both classified as urban areas (and thus being simulated with the same channel model), Zurich and Winterthur provide both the worst and the best coverage ratio, for any number of RSUs. The reason lies in the lower average vehicle speed and higher number of roads in the Zurich scenario, resulting in a longer travel time (compared to the simulation length) before reaching an RSU. This is confirmed by the delay between entering the map and receiving a beacon, shown in Figure 4.10(b), that is much higher for Zurich than for Winterthur. Concerning the suburban scenarios, Baar achieves slightly better performance than Baden because it has a lower number of roads, all crossing the same limited area; thus, a better coverage of the vehicles can be achieved. Figure 4.10(c) reports the CDF of the coverage time, when the number of RSUs is fixed to 6. The coverage time is computed, for each vehicle, as the sum of the intervals between back-to-back beacons received from the same RSU. Again, comparing this plot with the ones in Figures 4.6(a)-4.6(d), it can be seen that qualitatively similar results are obtained in the different road layouts. Also, note that the coverage times reflect the behavior of the coverage ratio shown in Figure 4.10(a); in particular, the probability of having zero coverage time is in agreement with the percentage of vehicles that do not receive any beacon, in all scenarios under study.

Next, the performance of content downloading is shown in Figures 4.11(a)-4.11(c). For each scenario, 1% of the total number of vehicles in the traces were selected to be downloaders. Their performance was monitored by setting τ equal to $1000\ s$ and placing a varying number of RSUs according to the MCTTP-g heuristic. Looking at the plots, it can be seen that results are affected by the road layout, the number of downloaders and the node mobility in the different scenarios. In particular, for a fixed number of RSUs that are deployed, a shorter total road length and a lower average vehicle speed result in a higher coverage time (Figure 4.11(a)), hence in a higher throughput (Figure 4.11(b)). This effect is especially evident by comparing the results obtained in the Baar area, where the total road length is shorter (hence RSUs are necessarily placed closer to each other) to those derived in the other suburban area, i.e., Baden, which is characterized by a larger total road length coupled with faster vehicles. Similar considerations hold for the comparison

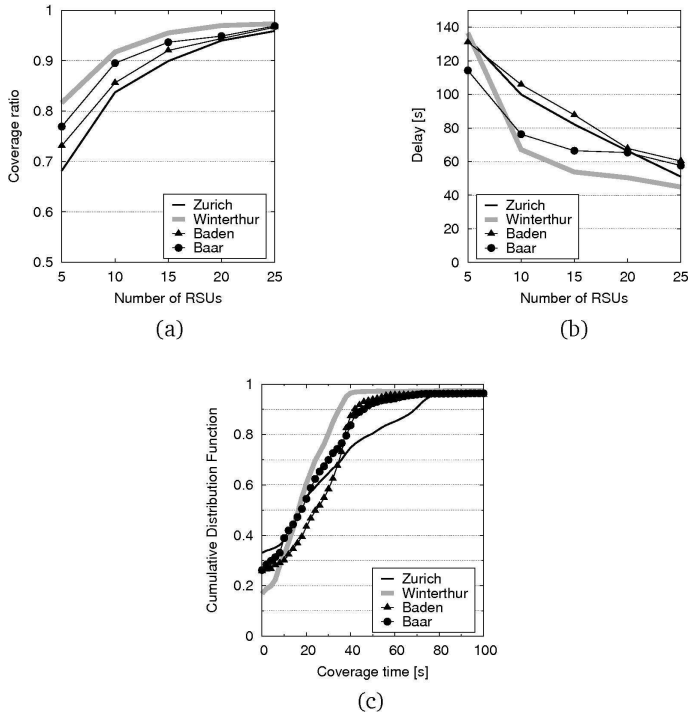


Figure 4.10: Dissemination case in the four scenarios: coverage ratio versus number of RSUs (a), delay between entering the map and receiving a beacon versus number of RSUs (b), CDF of the coverage time per vehicle with 6 deployed RSUs (c).

between the two urban scenarios, i.e., Zurich and Winterthur, where the former exhibits a larger road length than the latter. Another important factor in determining the performance is the number of vehicles under the same RSU that concurrently download information. The Baden and Zurich scenarios are the ones that exhibit a larger number of downloaders, and, consistently, feature a lower throughput than the others. It follows that the relationship between throughput and coverage time (the latter shown in Figure 4.11(b)) is not as strong as one would expect. Note also that the average per-downloader throughput is computed at the application layer, over the vehicles that are able to start downloading the file. An important metric to observe is therefore the percentage of downloaders that can retrieve not even a chunk, as well as the percentage of downloaders that is never under coverage of any RSU. Such results are reported in Table 4.1, for the various scenarios and as the number of deployed RSUs varies. It can be seen that the Baden and Baar areas, which are both suburban, exhibit better performance than the urban regions of Zurich and Winterthur. This is mainly due to the fact that in the former scenarios almost all vehicles travel on a few major roads, while in the urban environments downloaders may travel also on narrow roads with little RSU coverage. Furthermore, better performance is achieved in the Baar scenario than in the Baden area; indeed, the smaller number of roads, all within the same limited area, that characterizes the Baar layout leads to a larger coverage ratio.

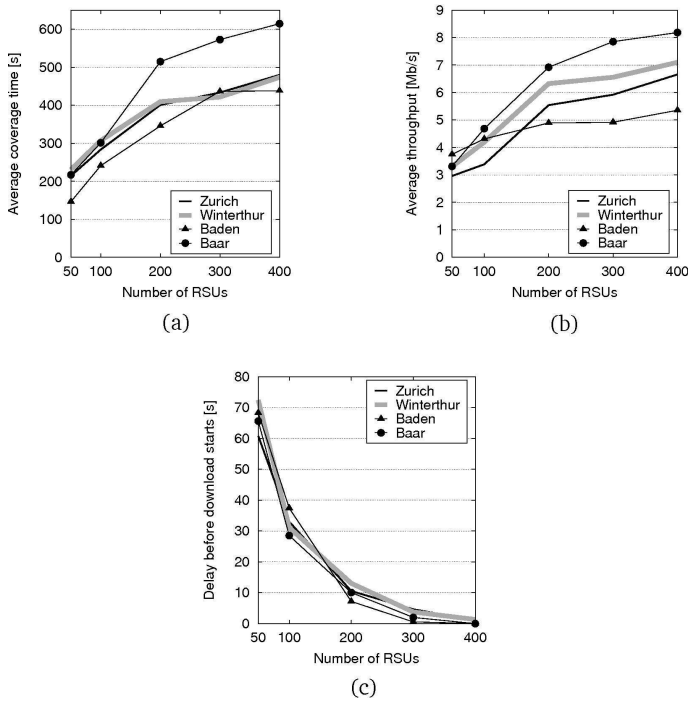


Figure 4.11: . Downloading case in the four scenarios: average coverage time (a), average throughput achieved by downloaders (b) and delay between entering the map and receiving the first chunk (c), versus number of RSUs.

Table 4.1: Content downloading: [percentage of downloaders experiencing zero throughput, percentage of downloaders experiencing zero coverage].

Scenario/No. RSUs	50	100	200	300	400
Zurich	[13.09, 12.99]	[12.4, 12.19]	[4.00, 2.00]	[2.00, 2.00]	[2.00, 2.00]
Winterthur	[12.50, 12.50]	[12.50,12.50]	[4.17, 4.17]	[2.09, 2.09]	[0, 0]
Baden	[15.75, 13.11]	[9.84, 9.84]	[4.84, 3.23]	[0, 0]	[0, 0]
Baar	[14.62, 12.50]	[3.12, 3.12]	[0, 0]	[0, 0]	[0, 0]

Finally, Figure 4.11(c) depicts the average delay between the time instant when the downloader enters the service area and the time at which it receives the first chunk (note that only downloaders that receive at least one chunk have been considered, in order to compute such a delay). Observe that the mobility scenario has little impact on such a metric, as the number of deployed RSUs is not as small as in the case of dissemination services; thus, it is likely that a downloader finds an RSU after a reasonable amount of time independently of the road layout. Also, as expected, the larger the number of RSUs, the lower the travel time before reaching an RSU, hence the experienced delay.

4.7 Conclusion

This chapter addressed two fundamental information delivery services in vehicular networks with infrastructure support: information dissemination from RSUs to passing-by vehicles and content downloading by vehicular users through nearby RSUs. In order to ensure good performance for both services, RSU deployment was investigated, by casting it as an optimization problem, and different formulations of the problem were presented.

Among such formulations, the one named Maximum Coverage with Time Threshold Problem (MCTTP) aims at guaranteeing that a large number of vehicles travel under the coverage of one or more RSUs for a sufficient amount of time, denoted by τ. Such a formulation also leverages, with respect to the others, the knowledge of the vehicular trajectories. The (either optimal or approximate) solution of the MCTTP problem emerged as the most suitable to support different information delivery services in vehicular networks, for a number of reasons. Firstly, it was shown that, by varying the minimum required coverage time, τ, in the MCTTP formulation, the RSU deployment can be calibrated so as to achieve a combined maximization of contacts and coverage time, which yields better performance than contacts-only or time-only driven solutions. Secondly, the MCTTP formulation leads to a more balanced distribution of the coverage time over the vehicles, with respect to other formulations: when RSUs are deployed according to the MCTTP solution, many vehicles experience a coverage time around the minimum required value threshold τ.

Simulation results, both under ideal and realistic conditions, also highlighted some interesting effects. A factor which is often underestimated, and that was investigated in this chapter, is the effect of the constraint on the coverage time. Simulation results showed that lower values of τ allow a fairer distribution of RSUs over the road topology, as more routes are covered, even if for a shorter time on average. On the contrary, increasing the τ threshold yields a more clustered RSU deployment, with the result that a smaller subset of routes enjoys a high coverage time, while the rest is left uncovered or almost so. We thus concluded that aiming at a given average coverage time does not necessarily mean that all vehicles will be under coverage of RSUs for so long, and fair coverage cannot be achieved. Furthermore, simulations allowed us to establish the impact of factors such as vehicle speed, vehicle density and road density, highlighting the need to collect these types of statistics when designing the coverage, for both information dissemination and content downloading.

In conclusion, RSU deployment cannot be addressed through random or intuitive placements, since neither one represents, in general, an optimal solution to the RSU deployment problem. Such simple strategies are easily outperformed by more sound approaches, for both the dissemination and the downloading objectives. The MCTTP formulation of the RSU deployment problem, which leverages the knowledge of the vehicles trajectories, represents an efficient solution and enjoys the desirable properties of (i) being configurable to a specific application by properly constraining the coverage time,

(ii) having an inexpensive greedy heuristics that well approximates the optimal solution, and (iii) yielding results that are consistently better than those achieved with unplanned RSU placements or Knapsack Problem-based formulations.

An important aspect to the performance of data delivery services that should be addressed by future research, and that is complimentary to the study presented in this chapter, is the interaction between the deployed RSUs and the vehicle-to-vehicle information transfer. Indeed, this chapter highlighted the unfairness in RSU coverage that may arise in realistic road layouts. A way to mitigate such an (often) unavoidable effect is to let vehicles share data of common interest in areas where connectivity with an RSU in not possible. If the amount of data exchanged through vehicle-to-vehicle communication could be predicted by the RSUs, an effective synergy could be created between roadside infrastructure and vehicles within the ITS.

Chapter 5

Optimization of Urban Traffic Flows

In this chapter, we propose an application that exploit the Vehicular Network to provide services to the users. Differently from previous chapters, here we address problems related to routing of vehicles instead of routing of information among vehicles. We suppose to have a system architecture enabling the exchange of information among vehicles, like one of those previously described. In the considered scenario, each vehicle communicates with a Central Controller using the available infrastructure, and relies on common navigation services. Exploiting the described architecture to collect the required traffic information, we propose a method to optimize urban traffic layout through basic heuristics and computationally efficient simulations. Instead of modeling an entire urban map with hundreds of intersections, each typology of intersection is simulated in order to understand how it responds to different traffic pattern and intensities. Then, this knowledge is leveraged to model the entire urban area and to compute minimal delay route for different flows of vehicles. The problem is studied by means of mathematical optimization. The proposed heuristic has been evaluated through simulations in realistic urban scenarios and compared with well-known existing routing strategies for vehicular traffic.

The content of this chapter is organized as follows. Section 5.1 introduces the optimization of urban traffic flow problem. Sections 5.2 and 5.3 describe the underlying idea and some possible applications. Section 5.4 explains the difficulty of representing urban traffic as a flow. Section 5.5 gives the details of our framework. Section 5.6 contains the experimental results, while section 5.7 discusses the previous works which are most related to our study. Finally, section 5.8 contains conclusions and future work.

5.1 Problem Statement

Traffic congestion is one of the major problems in urban areas. It impacts directly people's everyday lives and causes several billion dollars lost revenues each year [78]. Traffic congestion can at times be tackled by resizing roads and junctions so that they can serve more

vehicles. The roadway expansion problem reduces to classical bottleneck analysis which are widely exploited in computer systems and network design. Roadwork can then be planned empirically or as an optimization problem with respect to costs, time, space and user requirement [79]. Unfortunately, even when feasible, the increase of road capacity only mitigates the problem until the next inevitable demand growth. An alternative solution consists in efficient traffic control and management. Specifically, traffic control deals with tuning traffic lights, green waves and access ramp rates, so that the traffic density is sustainable for the infrastructure [80]. Traffic management proactively directs vehicles towards alternative routes in order to avoid jams [81]. Both approaches are driven by the goal of minimizing the delay perceived by the drivers.
Our work is inspired by computer network research that already dealt with congestion and optimization back in the days of the early Internet [82]. However, there is no satisfactory model for traffic flow on roadways as there is for computer network traffic. The complexity stems from the fact that vehicles do not operate independently but they constantly influence each other. Moreover, the problem is complicated by the unpredictability of the human behavior. The challenge addressed in this chapter is to understand the behavior of vehicular flows across urban intersections. In particular, this chapter clarifies how our urban traffic optimization has to deal with the following problems:

- The use of mathematical frameworks, such as linear or convex optimization, is not trivial because: (1) there is not a closed form solution to calculate the average delay of a vehicle on a route; and (2) the dimension of the optimization problem grows exponentially with the number of lanes and intersection that the system takes into consideration;

- It is computational expensive to reproduce traffic on a city-wide scale thus an optimal organization of traffic flows can hardly be found with an exhaustive search.

5.2 Contribution

The goal of this chapter is to find the optimal routing (i.e., minimum overall delay) in a urban grid that includes intersections. Tackling the problem of vehicular flow optimization as a whole is not feasible due to many variables. In the one hand, previous works show that the methods that account for intersection details do not provide a closed form solution required for efficient multi-commodity flow optimization. On the other, existing flow optimization algorithms are suitable only for low-density highway traffic, indeed without intersections. Henceforth this work formalizes how to break the problem into smaller tasks that can be completed using simple tools and basic algorithms.

The underline idea consists in categorizing road segments so that they can be simulated and modeled individually. First, the traffic behavior at different types of intersection under various conditions is derived, then this information is exploited to compute optimal

routing scheme for a given traffic demand. Based on this knowledge, the model of the system has been designed and used to minimize the average travel delay of the drivers.

This approach is promising for the following reasons:

- the approach does not depend on a specific simulator or traffic model. The choice of the simulator is intentionally left to the user as there are many simulators that can be used, but none that can successfully represents every scenario.
- In selecting the optimization algorithm it is possible to trade computational power for precision. Nevertheless, the results in this chapter show how a simple greedy algorithm can optimize traffic flows on a realistic urban scenario.
- The knowledge base is created just once and it is updated only if some intersections change. Note that in this case the new model of intersection can be simulated individually and the knowledge base can be updated incrementally. This is an important aspect since it might take days to simulate small urban area while it takes less than an hour to simulate a single intersection.

5.3 Example Scenario

The aim of this work is to accommodate the new generation of car navigators, in particular that ones that can receive remote instructions to use low congestion detours if the primary route is heavily congested. Today's GPS navigators take in consideration only distance and traffic updates. Unfortunately, this method does not work during rush hours because all vehicles on the same congested segment take the same detour causing route flapping [83]. The next generation navigator will assign drivers to different detours, reducing the probability of congestion. For example consider a rush hour situation when the same people must drive on the same routes from home from to work. Via a smart navigator, before leaving his garage, the driver is asked to choose the neighborhood closest to his destination. With that information, the Traffic Authority randomly assigns the vehicle to one of the possible routes that have been precomputed for the daily traffic demand. Note that this route might be longer than the straight path solution. However it has the advantage of being (1) decongested and (2) faster than traveling on the shortest path in case of traffic jam.

Feasibility of the Implementation

The whole idea of letting a computer (e.g., the navigator) deciding the route to the destination might sound excessive and even scary for the driver. However, the impact will be beneficial to society, considered that the probability that an individual is consistently routed toward longer routes is negligible. On average the miles driven will be about the

same than before but the total amount of time spent in the car will be much less. The main problem is the degree of penetration: this technology needs to be used by most of the drivers to be cost effective. To this aim it is possible to offer incentives to people who are willing to collaborate, for instance routes that are not suggested by the navigator for a particular driver are considered toll roads. Otherwise, people can be threatened with a fine if they do not respect their commitment. Note that it is not hard to verify if a vehicle is on its route, for instance it is possible to install at the main intersections RFID readers so that the presence of the vehicle can be recorded once it passes by. Viceversa, vehicles can send a beacon to base stations along the path to notify their presence.

5.4 Vehicles Behavior in a Urban Scenario

The goal of this chapter is to find the optimal routing (i.e. min overall delay) in an urban grid that includes intersections. For the sake of simplicity, only scenarios at full penetration rate with no selfish driver are considered in this work. A future extension will study the sensitivity of the approach to the different penetration rates. The previous works (see Section 5.7) indicates that the method that account for intersection details (e.g., cellular automata schemes) do not yield the closed form solution required for efficient multi-commodity flow optimization. On the other hand, existing flow optimization algorithms apply only to low-density highway traffic (no intersections). They cannot directly handle surface roads and residential scenarios. More specifically, conventional flow optimization methods cannot adequately approximate the impact of density impacts on: (1) the traveling time on each lane and (2) the waiting time at each intersection. Therefore the result is an oversimplified model that leads to misleading results.

In the following, a brief description of the vehicle interactions on the lanes and at the intersection is provided based on our analysis.

Vehicles on the Lanes

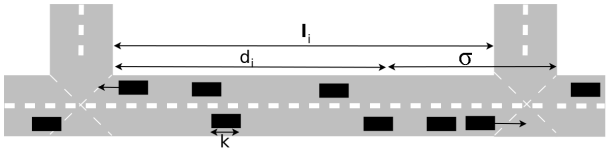


Figure 5.1: Interaction of vehicles in a road with one lane per-direction

Vehicles on the same lane influence each other by accelerating and decelerating as dictated by drivers habits. It is known, for example, that more aggressive drivers tend to maintain a smaller inter vehicle distance whereas safe drivers tend to slow down when approaching another car. Luckily, microscopic models such as Newell's car-following

[84] successfully capture all these aspects allowing to reproduce traffic dynamics such as *jams* and *shockwaves*. These phenomena cannot be noticed when considering vehicles as a flow. However, by taking into account their impact, on the flow model average delay, it is possible to anticipate them and slow down the traffic before severe congestion set in. Note that it is not feasible to redirect congested traffic in terms of flows. In case of congestion, vehicles must be rerouted individually and not as a flow. How vehicles interact on a single lane is shown in Fig. 5.1. In the picture, k is the vehicle length and I_i is the length of the lane. The road segment d_i is the space where normally vehicles are traveling whereas σ is the space where they start slowing down to approach the intersection. Accordingly to the majority of traffic models, vehicle interfere with each other by stretching the inter-vehicle distance s depending on the driver's aggressiveness. An aggressive driver tends to stay closer to the vehicle ahead of him whereas a safe driver slows down to keep more space in between. By slowing down he might force another vehicle behind him to slow down causing a chain reaction that propagates backward on the lane.

Vehicles in the Intersection

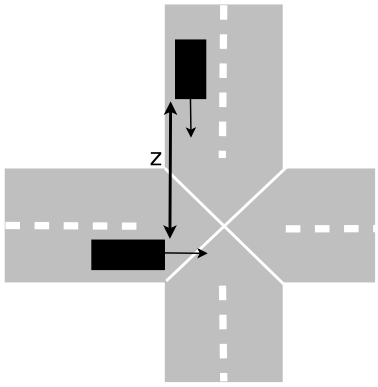


Figure 5.2: Interaction of vehicles at the intersection

Modeling intersections is even more complicated than modeling lanes, since vehicles interact in more than one dimension. Besides depending on drivers actions, the performance of an intersection is strongly influenced by:

- **Capacity.** Intersections have an implicit capacity bounded by the average time needed by a vehicle to move from one street to another. This value can be found with on-site measurements or can be computed from the average speed at the intersection. Note that, even knowing the capacity of an intersection, queue analysis still remains hardly applicable since the capacity is not uniformly distributed to each incoming lane. Instead, it varies with respect to factors such as geometry, policies, traffic density etc. Fig.5.2 shows that a vehicle crosses the intersection only if there

is at least a distance z from another incoming vehicle from a different lane. Note that, the distance z is not a constant instead it depends on the drivers attitude. An aggressive driver might cross the intersection forcing the vehicles with the right of way to slow down to avoid a collision. Fig. 5.5 shows some results about how intersection policies affect the capacity of each street lane. In Fig. 5.5(a) *flow 2* is the one with priority, while in Fig. 5.5(b) *flow 1* has the right of way against *flow 2*. Note how, even if in both the configurations there is only one flow with priority, the performances are not the same (in the first performances drop at $800\ v/h$, whereas in the other at 700). This is an example of how small factors can impact the performance of urban intersections In presence of a traffic light, the service time of the intersection is time shared among all the lanes in such a way that these almost never interfere with each other. In this particular case convex optimization can be used to optimize the traffic demand.

- **Enviroment.** The environment influences the way drivers approach the intersection and consequently its overall performance. This aspect can be better observed with real measurements rather than simulation although factors such as narrow roads, blind spots, bicycle lanes or pedestrian crossing impact sensibly the performance of the intersection.

- **Policies.** The delay experienced by vehicles at an intersection strongly depends on the policy of the intersection, e.g., right-before left, no right turn on red, traffic lights etc., and the amount of vehicles traveling on each incoming lane. When streets with priority are subject to high traffic volumes all the other incident streets become congested because the high priority streets are capturing the entire capacity of the intersection.

5.5 Model

This model has been developed by assuming that waiting time at an intersection is much smaller than the traveling time on the lanes between two intersections. However, if one or more intersections are jammed, the intersection becomes the bottleneck and the time spent on the lanes is of secondary importance. With this key assumption the following model has been formulated. A urban map is considered as $M = (I, L)$, where I is the set of the intersections and L is the set of lanes on the map. Each lane $l_i \in L$ is assumed to have (1) known length d_i and (2) a known vehicle average speed v_i. Traffic demand is expressed as:

1. A set of users $U = \{u_k : u_k \in I \times I\}$. Because flows are considered from intersection to intersection.

2. A function $\Phi(x) : U \mapsto \mathbb{R}$ that maps users to their $vehicles/hour$ demand.

Each user can be served by a number of routes from a set $R = \{r_i : r_i \subset L\}$. From now on, $r_i \in u_k$ signifies that route r_i serves user u_k. Analogously $l_i \in r_j$ means that route r_j contains the lane l_i. Two vectors $\mathbf{f} = \{f_1, f_2, ...f_{|L|}\}$ and $\mathbf{y} = \{y_1, y_2, ..., y_{|R|}\}$ represent the current state of the system, f_i contains the amount of flow on lane l_i whereas y_j contains the amount of flow on the route r_j.

By design, this model does not handle heavy, congested traffic as the latter cannot be properly represented as a continuous steady state flow. However, if applied incrementally, this approach delays the congestion build-up until the capacity of the road system is reached.

Formally, it works under the assumption that, given a lane l_i, vehicles never stop before having driven a distance $d_i - \sigma$ on it, where σ is the space of the lane used by vehicles to decelerate and approach the intersection. The reason is that given a lane l_i the time spent on it, i.e., $T_i(\mathbf{f})$, must be split into two components: the amount spent at the intersection w_i and the time spent moving on the street segment l_i. This leads to the expression:

$$T_i(\mathbf{f}) = \frac{d_i - \sigma}{v_i} + w_i(\mathbf{f}) \tag{5.1}$$

Thus, the total time spent on a route r_j is equal to:

$$T(r_j, \mathbf{f}) = \sum_{l_i \in r_j} T_i(\mathbf{f}) \tag{5.2}$$

Thus, the minimum average delay for the drivers can be written as:

$$\min_{\mathbf{f}} \left(\frac{1}{\gamma} \sum_{l_i \in L} T_i(\mathbf{f}) \cdot f_i \right) \tag{5.3}$$

where γ is equal to $\sum_{r_j \in R} y_j$. Note that, if this function were convex the problem could be solved with the flow deviation method or similar frameworks. Unfortunately, as the results in Fig. 5.6 show, the function does not remain convex due to $w_i(\mathbf{f})$. The time spent at an intersection depends on the load of each incident lane and, to the best of our knowledge, beside [85] there is no previous work for a synthetic representation of the different types of urban intersections. In order to complete the task, our model considers the time spent at an intersection with n incident lanes as a generic function $\mathbb{R}^n \to \mathbb{R}^n$ where the domain is the amount of flows on the incident lanes and the codomain is the delay on each lane. An estimate can be obtained by simulation.

Simulation

Before running the simulations, intersections must be classified with respect to their geometry. This task can be easily done via software even though some urban areas can

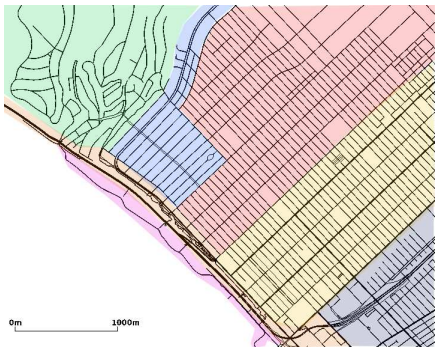


Figure 5.3: Example of recognizable urban patterns in Santa Monica (CA).

simply be visually inspected. For example, in Fig. 5.3, it is clear which intersections have the same geometry and which do not. Note the symmetry of the areas in red, yellow, purple and blu. By zooming in, intersections can be recognized to have the same geometry. Differently, in the area colored in green each intersection is different from the others and clearly need to be inspected further. The process of deriving a delay function can be summarized as follows:

1. The process starts with an initial minimum flow on each lane. Then, at each iteration one of the flow is incremented of a given step until all the lanes are saturated. The smaller is the step, the more precise is the resulting function.

2. Simulation run so that both the time a vehicle arrives at a distance σ from the intersection and the time the vehicle enters into another lane are recorded. The difference between the two is then averaged with basic statistic.

It is important to verify that vehicles depart at random instants of time before approaching the intersection. The reader is owed of an explanation of why simulations are ran only on intersections and not on the entire lanes. In particular:

- The performance of streets can be usually approximated dividing their length by the average speed. Intersections, instead, follow complex dynamics that do not have a closed form solution. Although, on the other hand, intersections with similar geometry, and same strategy, show similar performance, as in [86]. This chapter aims at minimizing time and costs of traffic management by reducing the number of simulations needed (simulation requires expensive hardware for computation).

- If there is no traffic jam, the time that a car travels before approaching the intersection is quite accurate.

- It is relatively easy to categorize intersections on a map by inspecting their geometry.

Algorithm

At last an heuristic must be chosen to compute optimal routes. As a matter of fact, there are many algorithms that can be used, each one with a different tradeoff between performance and precision. Here we present a very intuitive greedy algorithm that helped us to prove the validity of our assumptions and gave good results in practice, as shown in Fig. 5.4. The algorithm simply loops over each user u_k and distributes a fraction δ of its traffic demand $\Phi(u_k)$ on the route that would increase the total average delay the least. This can be formalized as:

$$opt(u_k, \mathbf{f}) = r_i \;\; s.t. \;\; \frac{\partial T(\mathbf{f})}{dx_i} = \min_{j \in u_k} \left(\frac{\partial T(\mathbf{f})}{dx_j} \right)$$

The pseudo-code is the following:

Algorithm 1 Greedy Algorithm

1. while $tot < \gamma$
2. $\mathbf{y} \leftarrow [0,0,...,0]$
3. $\mathbf{p} \leftarrow [0,0,...,0]$
4. for each $u_k \in U$
5. $min = opt(u_k, \mathbf{f})$
6. if $(p[u_k] < \Phi[u_k])$
7. $y[min] = y[min] + \delta$
8. $p[u_k] = p[u_k] + \delta$
9. $tot = tot + \delta$

The time complexity is $O(U)$; however, as most of the greedy algorithm, this is an approximated algorithm which can deviate from the optimal solution. A theoretical bound to the quality of the approximation is part of the future work, while the experimental results can be found in the next section. In any case, the approach described throughout this chapter can be used with any other heuristic that can optimize the objective function (5.3).

5.6 Evaluation

For the experiments we used the open source simulator SUMO (Simulation of Urban MObility) [87]. Unfortunately, due to the complexity of parsing SUMO map files, experiments could be run only over Manhattan grids with 11×11, 21×21 and 31×31 intersections of the same type. Future work will include real urban maps as the one in Fig. 5.3. Each road segment on the grid is $400\ m$ long and vehicles travel on it with an average speed of $60\ Km/h$. Traffic demand consists of two orthogonal flows moving respectively from East to West and from North to South.

The experiments used two different kinds of intersections:

- *Priority*: two parallel lanes out of four have absolute priority on the others; this implies that parallel lanes do not interfere with each other. Also, when the priority lanes reach a certain density, the other two cannot use the intersection anymore, as shown in Fig. 5.6.

- *Right before left*: This intersection is the most interesting because all the four lanes impact, directly or indirectly, on the others.

Vehicles moved on the map using the car-following model as described in [88].

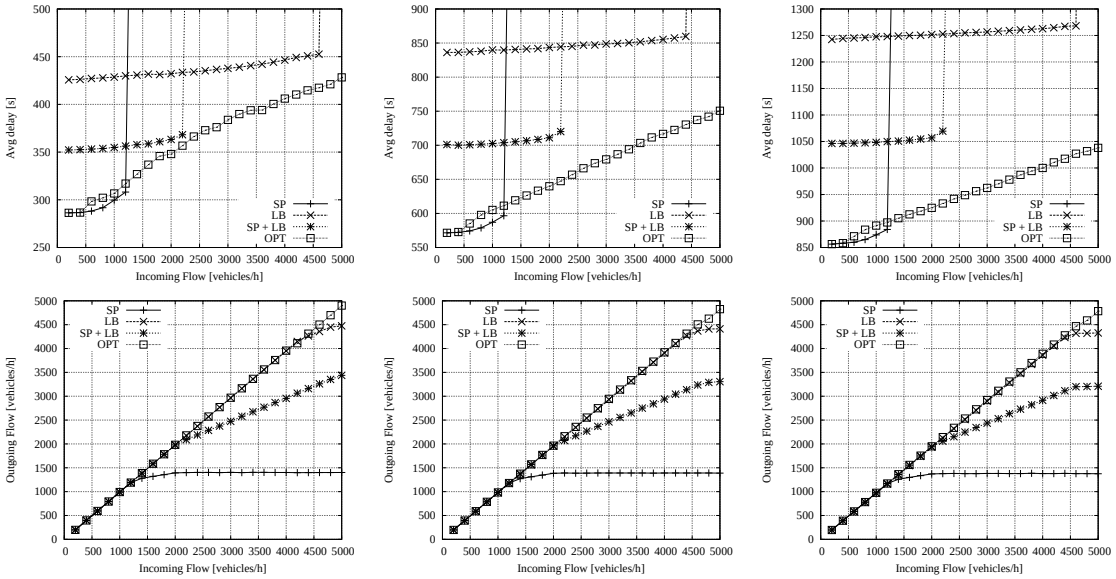
Comparison

The proposed greedy scheme has been compared against three different routing policies:

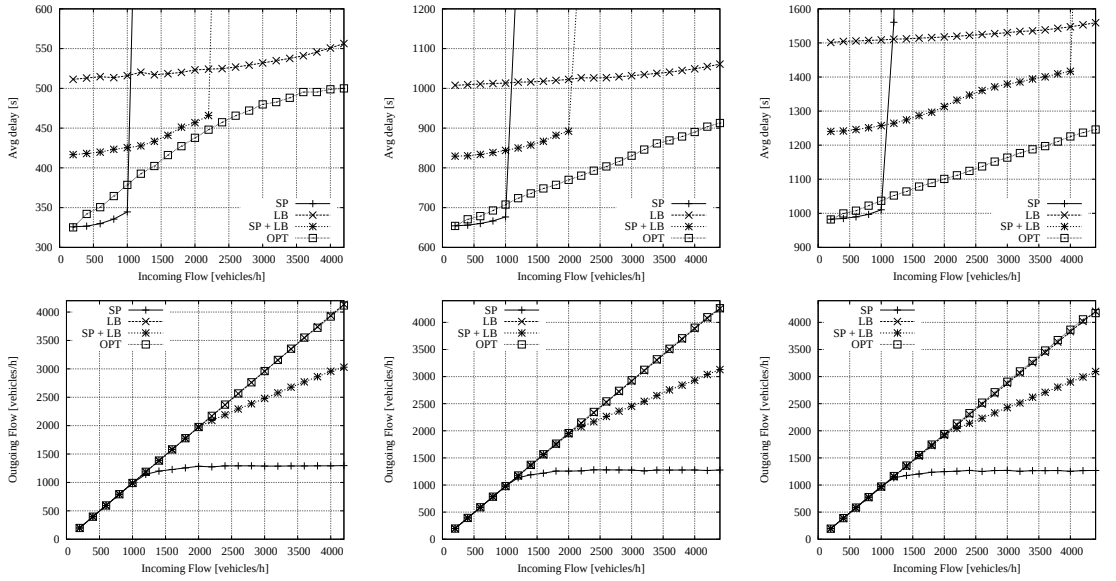
- *Shortest Path (SP)*: traffic demand is directed toward its shortest route. Obviously, this policy initially leads to the minimum average delay, but it causes congestion earlier. Hence, it has been used as a lower bound.

- *Load Balancing (LB)*: traffic demand is equally split on its possible routes. Note that, on a Manhattan grid, load balancing would be the routing policy that delays congestion the most if traffic could move as a simple fluid.

- *Mixed Strategy (SP+LB)*: this policy combines the two previous approaches. The flow moving from North to South is directed on the shortest path, whereas the other is equally divided on all the available routes.



(a) Priority



(b) Right before Left

Figure 5.4: The columns show the results obtained on three Manhattan grids, respectively with (from left to right) 11×11, 21×21, and 31×31 intersections per side. Graphs (a) refer to a map with intersections of type *priority* while (b) refer to *right before left*. The greedy method (OPT) is compared against: *shortest path SP*, *load balancing LB* and *mixed strategy SP + LB*.

Results

The graphs in Fig.5.4 show: (1) the average delay of the routes computed with OPT is overall better than the others. Note that for low demands OPT does not follow the shortest path because of the drift caused by approximating the cost function of each lane. If

simulations were run with a smaller step OPT and SP would have the same trajectory. (2) OPT delays congestion more than any other routing scheme. (3) Even when routes are congested, OPT still moves more vehicles/hour than the other algorithms. It is clear that, even by taking into account only the delay introduced by the intersections, the results produced by our "greedy" optimization far outperform the typical load balance and shortest path strategies. This is due, in great part, to the fact that our model, albeit approximate, still reproduces the critical vehicle-to-vehicle interactions in a realistic way. Eventually, also the greedy approach will lead to congestion for large enough offered loads. One may argue that better heuristic solutions could be obtained by recalculating the entire traffic layout after each increment in demand. Obviously, the time complexity of the latter method would be much higher. A possible compromise is to adopt a randomized approach to search for local minima, an option that we are currently investigating.

5.7 Related Work

Here, we discuss previous work dealing with traffic models and traffic management which are mostly related to our study.

Traffic Models

Mobility models are classified under three categories, as by Fiore et al in [89], that offer different tradeoffs between performance and complexity: Macroscopic, Mesoscopic and Microscopic.

Macroscopic models look at traffic as a continuous flow of vehicles. This is the highest level of abstraction and it is hardly applicable to an urban scenario that has intersections and cross traffic. The aim of this chapter is to use this level of abstraction to optimize delays in a urban scenario without losing precision. Noticeable contributions are:

- From Prigogine et al. [90]. This model investigates the interactions of two traffic flows in a urban scenario. It is a valuable contribution although it is quite far from modeling realistic intersections.
- Fluid Traffic Model (FTM) [91]. This model adapts the speed of vehicles according to traffic density. It considers a single lane without accounting for multi-flow interactions.

Microscopic models consider vehicles individually and capture the effect of their interactions. This is the finest grain that can be used to simulate vehicles dynamics but it requires high computational power and system memory. Given the level of detail, this approach cannot scale to represent an entire neighborhood or a whole city. Popular examples of discrete microscopic models are cellular automata models that are quite often used for larger scales. In particular:

- Car following models. In this kind of models vehicle motion is described by Ordinary Differential Equations (ODE) or a Algebraic Differential Equations (ADE). Popular examples are the Intelligent Driver Model (IDM), the GHR Model [92] and the Krauss Model [88].

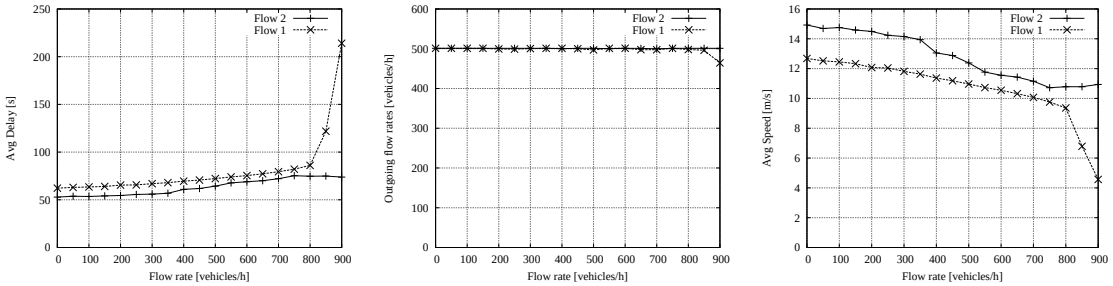
- Nagel and Schreckenberg [93] is a stochastic discrete automaton model to simulate freeway traffic. This work consists of a discrete model that allows to represents important phenomena such as (1) traffic jam and (2) the transition from laminar to start-stop-traffic.

- Biham et al. in [94] describes how traffic flows interact on a two dimensional space. In spite of the simplicity of the model, the results show a sharp transition that separates a low-density dynamical phase in which all cars move at a maximum speed and a high-density static phase in which they get stuck in a global traffic jam.

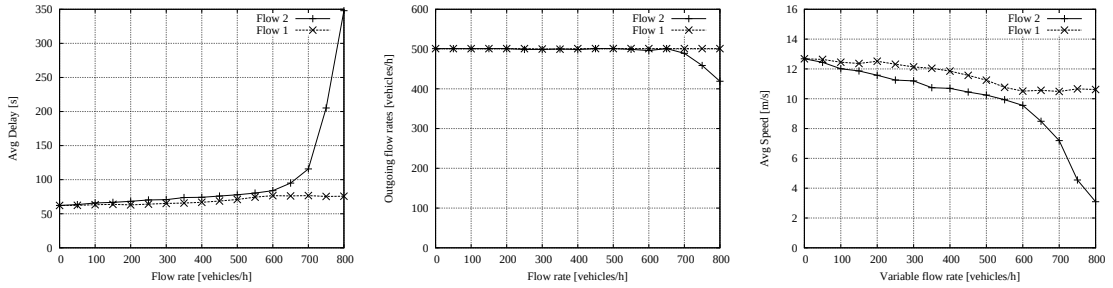
Mesoscopic models compromise between the simplicity of a macroscopic approach and the precision of microscopics. Vehicles usually move in groups or clusters (e.g., platoons) so that the probability of a specific vehicle being in a certain place at a specific time instant can be bounded. This solution offers a tradeoff between car level precision and the complexity of microscopic models, although it can deviate significantly from the real scenario.

Traffic Management

Reactive schemes. They are usually studied from the control theory point of view: traffic is seen as a dynamic system. A feedback mechanism triggers a controller to tune the traffic regulators (e.g., traffic lights, access ramp controls, etc). In [95], two schemes are used: a local balancing scheme and a global scheme. The global scheme assumes global traffic knowledge at each node, which is unlikely to happen in a real environment. Moreover, in the latter case the routing is done by each vehicle independently and thus it is subject to route flapping. The local scheme, on the other hand, assumes that the On Board Navigator has heard from its peers in the adjacent road segments and uses that information to perform local route balancing. Results in [95] state that Global Optimization is not any better than local load balancing. However, these results must be taken with a grain of salt since they assume that the traffic is randomly distributed. This implies that there are no major hot spots, such as a congested freeway access ramp, which is unlikely. In [96] Mohandas et al. propose to use the Adaptive Proportional Integral rate controller as it has been done for the Internet to deal with links congestion. The work showed the applicability of the method although there has been no experimentation in the urban scenario. Likewise, in [97] there is an extensive theory for vehicular traffic control without attempts to apply it to a real world scenario.
Proactive schemes. They consist in preemptively computing a best solution for traffic



(a) Priority



(b) Right before Left

Figure 5.5: The graphs show the performance of two kinds of intersections: (a) *priority* and (b) *right before left*. Each row shows respectively (from left to right): delay, outgoing flow rate, and average speed. Each graph plots the performance of a flow whose intensity varies (on the x-axis) against a constant flow set to $500\ vehicles/hour$.

using a knowledge base created with real measurements or via modeling and simulation. For example, Liu et al. in [98] use historical records to calculate the optimal routes for the drivers. Other proposals, such as [99], suggests the use of bayesian networks to infer traffic dynamics.

Flow-based schemes. Many proposals are inspired by previous work on computer networks optimization, such as [81] [100] [101]. For instance, Kim et al. in [81] use the Flow Deviation algorithm for load balancing traffic demand over alternative routes. This model has appeal because it leads to a closed form expression of delay as a function of road segment parameters, and can be used to obtain optimal routes using convex optimization. Lanes act as the links of a network and intersections act as the routers. Each lane l_i has a maximum capacity C_i, measured in $vehicles/hour$, and a traveling time T_p (analogous to the propagation time of networks links) measured in seconds; intuitively vehicles move from link to link like packets do in a computer network. From [82], the average delay of a vehicle is equal to the time spent in a M/M/1 queue with service rate $\mu_i = C_i$ and arrival rate λ_i equal to the intensity of incoming traffic on the lane. The total delay on the lane has two components: $T_d = T_p + T_q$ where $T_q = \frac{1}{C_i - \lambda_i}$ is the average queueing time spent at the ending intersection of lane l_i and $T_p = \frac{Lane\ Length}{Average\ Speed}$ is the average traveling time on the lane. The average delay perceived by vehicles can then be

rewritten as:

$$T = \sum_{l_i \in L}^{n} \frac{\lambda_i}{\gamma} T_d(i) \tag{5.4}$$

where n is the number of lanes in the map, γ is the total vehicle arrival rate to the system and $T_d(i)$ is the average delay of lane l_i. Unfortunately, vehicles interact in a different way than packets thus the result that can be obtained with this model are quite far from what can be observed in reality. For example, packets in a router can be switched from input to output port with minimal overhead, while vehicles suffer the most severe delays at intersections. These models are based on the unrealistic assumption that a road lane can be approximated with basic queues, such as an M/M/1. Other works, such as [102], dug more into advanced queueing theory for a finer representation of the problem. Unfortunately, while simple queue models fail in describing the real dynamics of vehicles, the advanced queue models are too complicated to be applied to a real world scenario. Instead, our chapter proposes an hybrid approach that leads to realistic results using relatively simple algorithms and inexpensive simulators.

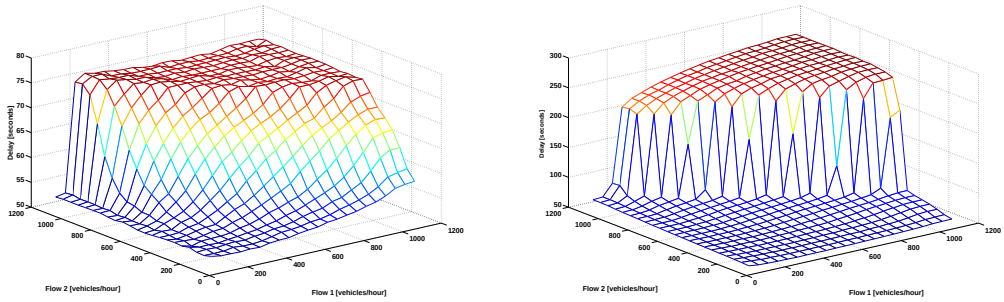


Figure 5.6: The first graph (left) shows the delay experienced by the vehicles traveling on Flow 1 which has not priority at the intersection. The second graph (right) plots the delay experienced on Flow 2 which has the priority. As it can be seen, the allocation of capacity is not easy to represent with a simple convex function.

5.8 Conclusion

In this chapter we described an efficient yet simple way of optimizing traffic flows on a realistic urban scenario. The proposed solution is computationally efficient, accurate and inexpensive to be deployed. As future work, we would like to optimize traffic on a real map and to study the algorithm sensitivity for different penetration rates.

Chapter 6

Routing Strategies for Electric Vehicles

In this chapter, like in the previous one, we present an application that exploit the Vehicular Networks to efficiently route vehicles in a urban area. In particular, here we focus on the problem of Electric Vehicles drivers' assistance through ITS. Drivers of EVs that are low in battery may ask a navigation service for advice on which charging station to use and which route to take. A rational driver will follow the received advice, provided there is no alternative choice that lets the driver reach its destination in a shorter time, i.e., in game-theory terms, if such advice corresponds to a Nash-equilibrium strategy. Therefore, we solve the problem using a game-theoretic approach, envisioning two models, namely a congestion game and a game with congestion-averse utilities, both admitting at least one pure-strategy Nash equilibrium. Using our models, we show that the average per-EV trip time yielded by the Nash equilibria is very close to the one attained by solving a centralized optimization problem that minimizes such a quantity. The proposed models have been evaluated both through mathematical analysis and simulations in realistic urban scenarios.

The content of this chapter is organized as follows. In Section 6.1 we introduce the problem of Electric Vehicles drivers' assistance through ITS, while in Section 6.2 we discuss previous work highlighting the novelty of our contribution. The system scenario is introduced in Section 6.3, along with the statement of the problem under study. We motivate our work in Section 6.4, by showing that centralized optimal solutions may lead to advice that may not be followed by the EV drivers. The game-theoretic approach that we adopt for the problem solution can be found in Section 6.5. In Section 6.6, we introduce the simulation scenario that we use to derive the results presented in Section 6.7. There, we show the low complexity of the proposed method and its benefits in terms of per-EV trip time. The latter results are derived through the SUMO simulator [87] and using a real-world road topology. We draw our conclusions and discuss future work in Section 6.8.

6.1 Problem Statement

Any technology touted as environmentally-friendly is likely to have its place secured on news media around the globe. Among green solutions, Electric Vehicles (EVs), viewed by all as emission-free, clean and noiseless, are rapidly rising in popularity and expectations, also thanks to the predictable shortage of fossil fuel in the not-so-distant future. Indeed, EV mass-production and widespread adoption seem all but likely if some early hurdles are overcome, such as short driving range, lack of refueling (i.e., charging) infrastructure and long charging time.

Arguably, any road scenario in ten years' time will likely feature some ratio of EVs taking over the streets. Old-fashioned gas pumps might also be gradually phased out by public charging stations, with electric outlets popping up in places such as curbside parking, parking lots and cab stands.

Even in this rosy scenario, one wonders when worries about vehicle range and availability of charging stations will be lifted and whether drivers will not be forced to plan their entire trip or commute around such availability, at least early on in charging station development. Finally, it is not clear when the "time consuming" tag will be removed from the task of car recharging.

Given the above concerns, ICT and ITS (Intelligent Transportation Systems) can step in and provide solutions that alleviate such concerns. Indeed, traditional navigation services could be integrated with the information provided by roadside network infrastructure and on-board user terminals through wireless communication [103, 104]. A Central Controller (CC) could collect information on the vehicular traffic conditions and on the occupancy status of the charging stations through ITS facilities. Then, EV drivers that need to recharge their batteries could send a request to the CC and ask for advice on the specific charging station to choose and the route to take.

The key point in this scenario, however, is that drivers that resort to such a navigation service will very likely behave as self-interested users, who aim at reaching their destination in the shortest possible time (including the time they have to stop at the charging station). Thus, they will follow the advice provided by the CC only if they find it convenient to themselves.

In this work, we show that the advice provided by the CC may not conform to the interests of EV drivers when it is obtained by solving a centralized optimization problem that, e.g., minimizes the average per-EV trip time or the maximum EV expected trip time. We demonstrate instead that the above requirement is satisfied when the problem is modeled as a non-cooperative game. Specifically, we resort to a congestion game [105] and a game with congestion-averse utilities [106], where the players are the EVs that need to recharge their batteries. In such games, the decision to be made concerns the charging station that an EV should use, along with the route to take passing through such a station. The two game models exhibit a different level of realism and complexity; however, for both of them, we show that, when the CC uses the game solution to provide advice to the

EVs, the following facts hold.

(i) The navigation strategies suggested by the CC correspond to Nash Equilibrium (NE) strategy profiles[1], i.e., each EV finds the suggestion by the CC convenient to itself and is willing to adhere to it.

(ii) The advice provided by the CC leads to an average per-EV trip time that is very close to the minimum obtained by solving a centralized optimization problem, and much shorter than the one EV drivers can obtain by adopting a greedy approach (e.g., always select the closest or the least congested charging station). This is highly desirable since, shortening the average per-EV trip time, contributes to reducing road congestion and energy consumption due to EVs.

6.2 Related Work

Recently, both the academic and industrial communities have devoted a great deal of interest to EVs and to the use of ITS services in support of EV drivers.

As an example, in [107] Ferreira et al. consider the case where the behavior of EV drivers, i.e., whether they drive to the closest or the cheapest charging station, depends on their profile (age or gender). The authors design a system that, through various communication technologies, provides EV drivers with several information, among which, the locations of the charging stations. The burden of selecting the charging station, however, is left to the drivers, as the study of the trip time associated to different alternatives is not within the scope of [107].

An analytical model for the study of the EVs trip time is presented in [108]. The road topology is modeled as a graph whose edges are associated with a fixed, i.e., non traffic-dependent, waiting time. Charging stations are likened to multi-server queues, and a theoretical lower bound to the charging time is derived. The model, however, does not include the availability of a central controller and, unlike our study, it does not consider that vehicles may deviate from their originally-planned route in order to reach a suitable charging station. Thus, the study in [108] does not account for the EV travel time to and from a charging station.

The works in [109–111] are mostly concerned with the EV consumption and its impact on the power grid. In particular, in [111] the authors envision a central controller that predicts the EVs mobility and advises each EV about which charging station to use and when, so as to smooth the power consumption peak. The work in [111], however, accounts neither for the time the EVs may have to wait in line at the charging station, nor for the

[1]Recall that an NE is a game solution, in which each player is assumed to know the equilibrium strategies of the other players, and no player can gain anything by unilaterally changing his own strategy.

fact that EVs may act strategically. The study in [112], instead, focuses on estimating the battery discharge time. The trips of the EVs are modeled using real data and traffic statistics, and vehicles are assumed to use the closest available charging station. Again, the waiting time at the charging station and the fact that EV drivers may significantly deviate from their planned route to reach a station are neglected.

At last, we mention that in [113], we presented a preliminary work that investigates *which* information is important that EV drivers receive through ITS. In particular, we showed the benefit of transmitting specific suggestions to EV drivers on which charging station to use with respect to the case where only mere updates on traffic conditions and charging station occupancy are provided. The evident advantages brought by specific advice motivated our present work, which is concerned about *how* such advice should be determined.

6.3 System Scenario and Problem Statement

We consider a road topology including a set of *road segments* $\mathcal{L}$ and a set of *charging stations* $\mathcal{C}$. Any ordered sequence of adjacent segments $l \in \mathcal{L}$ is said to form a *route*.

Among all vehicles that travel across the topology, we identify the following three categories:

- non-EVs or EVs with medium-high battery level, which are not interested in using a charging station;
- EVs whose battery is low, but that will not resort to the navigation service to identify the charging station;
- EVs with low battery that use the navigation service to select a charging station. If they find it convenient, they may deviate from their original route to reach a charging station.

Note that the vehicles in the first category just contribute to the traffic intensity over the roads, while those in the last two categories contribute both to the intensity of vehicular traffic and to the occupancy of the charging stations.

For the vehicles that stop at a charging station, it is fair to assume that their battery is replaced with a fully-charged one. Only in the unlikely case where no one is available, is the EV battery recharged. This choice is due to the charging times approaching half an hour, according to today's fast recharge technology [114]. Charging stations may have a number of replacing stalls (hereinafter servers), possibly varying from one station to another. Clearly, upon reaching a charging station, an EV will incur a waiting time that depends on the occupancy of the station, the service time and the number of available servers.

Next, we focus on EV drivers that belong to the last category, i.e., they have got a low battery level and resort to the navigation service. As mentioned, such EVs can be considered as self-interested users. Specifically, we assume that *their goal is to minimize the total trip time* toward their intended destination. This translates into assuming that drivers consistently act in order to pursue such an objective – as opposed to, e.g., driving to the charging station they like better, or to the one where they can collect bonus points or miles.

In the most general case, such EV drivers may be able to reach a number of possible charging stations and, for each of them, they may choose among multiple, different routes. Therefore, they will ask the advice of the CC to make a decision on the charging station to use and the route to take, including their current position and final destination in the request. It is fair to assume that the CC has knowledge of the road topology and traffic conditions, as well as of the locations of the charging stations, their current occupancy and availability of fully-charged batteries. Based on such information, the CC indicates to the EVs which station to use and the route to take. Upon receiving a response from the CC, all rational, self-interested EVs that made a request will be willing to follow the CC's suggestion if this conforms to their own interest. Some EV drivers, however, may not be rational and eventually decide not to adhere to the received advice; we will take this into account in Section 6.7.

6.4 Why a Game Model?

A natural choice to solve the problem of selecting the charging station for each EV, and the corresponding route, would be to let the CC formulate an optimization problem that minimizes the average per-EV trip time. This would be desirable as it would lead to reduced road congestion and to energy savings, i.e., to maximizing the social utility of the system. However, it is easy to show that in general such an approach yields solutions that EV drivers may find not convenient to themselves, hence to which they will not adhere. The same observation holds in the case where the CC tries to minimize the maximum EV expected trip time.

As an example, consider the toy scenario depicted in Fig. 6.1, where there are two charging stations, c_a and c_b, both with one idle server and service time equal to 2 time units. Assume that, at the same time, two EVs, v_1 and v_2, have low battery and ask for the help of the navigation service to select the charging station to use. EV v_1 can reach either c_a or c_b, but its travel time toward the two stations is 2 and 1 time units, respectively, while from both stations to its final destination, d_1, the travel time is equal to 1 time unit. EV v_2 instead can only head toward c_b, with travel time equal to 1.5 time units, and from there it can reach its destination d_2 in 1 time unit.

It is easy to verify that, if the CC provides its advice to the EVs so as to minimize either the average per-EV trip time or the maximum EV expected trip time, then the solution is:

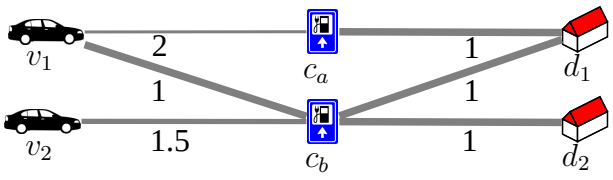


Figure 6.1: A toy scenario.

v_1 heads to c_a while v_2 uses c_b. This would indeed lead to an average per-EV trip time and a maximum EV expected trip time equal to 4.75 and 5 time units, respectively[2]. However, v_1 will not find the CC's advice convenient to itself as, by heading to c_b, it would incur a total trip time of 4 time units, against the trip time of 5 time units it would experience by following the CC's suggestion. Thus, v_1 will ignore it.

Based on the above observation, we propose a different approach. We model the problem of selecting the charging station, and the corresponding route, as a non-cooperative game, in which the players are the EV drivers that resort to the navigation system for advice. Then, we look for a strategy profile that is an NE and is convenient from the viewpoint of the system performance, and we take this as a solution to the problem. Since in this case the advice by the CC corresponds to an NE, there is no alternative choice for an EV that leads to a shorter time to destination, hence self-interested drivers will adhere to it. For instance, in the example above, the CC will suggest to both v_1 and v_2 to use c_b, and no one will deviate from the CC's advice.

It is clear, however, that a game-theoretic approach does not ensure that the average per-EV trip time[3] is minimized (e.g., in the above toy scenario it increases from 4.75 to 5 time units). Nevertheless, in Section 6.7 we show that, even in real-word scenarios, the average per-EV trip time obtained through our game-theoretic approach is remarkably close to the optimum.

Finally, it is important to stress that the game could be solved by the EVs themselves, provided that they have the required information. In our case, however, we take a practical perspective and consider that it is the CC that collects all the information, processes it and solves the game so as to provide the EV drivers with the strategy to adopt (i.e., the charging station to use and the route to take). This implies that the proposed mechanism neither significantly increases the system overhead due to communication protocols, nor requires EV drivers to exchange sensitive information about themselves, or make any computation to make a decision.

[2]If v_1 uses c_a, its trip time is 2+2+1=5 time units, while v_2's trip time is 1.5+2+1=4.5 units. This results in an average per-EV trip time of 4.75 and a maximum EV expected trip time of 5. If instead v_1 heads toward c_b, it arrives there first and its trip time becomes 1+2+1=4 units, while v_2 finds the station server occupied by v_1, thus its trip time increases to 1.5+1.5+2+1=6. It follows that the average per-EV trip time and the maximum EV expected trip time become 5 and 6 time units, respectively.

[3]In game-theoretic terms, this quantity is called price of anarchy (PoA).

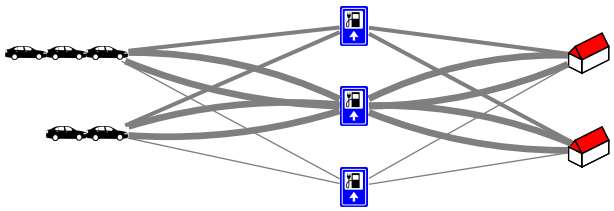


Figure 6.2: Abstract representation of the vehicular scenario where each EV may take several different routes to a given charging station and from there to its intended destination.

6.5 The Recharging Game

We now detail the game models we use to solve the recharging problem in the system scenario described in Section 6.3.

Assume that the CC processes the requests received from EVs with low battery every T seconds. We denote the set of EVs that ask for advice during a T-second time period by $\mathcal{N}$, and its cardinality by N. Consider the most general case in which each of the N EVs may reach several charging stations and take different routes to arrive at a given station, as well as to go from there to its final destination. For clarity, we depict an abstract representation of such a scenario in Fig. 6.2; we will deal with a real-world road topology and realistic vehicular mobility while deriving the performance results in Section 6.7.

In the figure, lines connecting vehicles with charging stations, and the latter with final destinations, represent the possible road segments that EVs can take to or from the charging station. The different thickness of the lines denotes the fact that road segments may be characterized by various levels of traffic intensity, hence they may imply different travel times. Clearly, in a more general setting, road segments may end at any intersection on the map, other than a charging station or an entry/destination point.

We then consider the N EVs to be the players of a congestion game [105], i.e., a non-cooperative game, in which players strategically choose from a set of facilities and derive utilities that depend (in an arbitrary way) on the congestion level of each facility, i.e., on the number of players using it. Congestion games are of particular interest to us since they have been proved [105] to admit at least one pure-strategy[4] NE. It follows that, if the CC derives its advice by modeling the system as a congestion game and finding a solution that is an NE, then all rational, self-interested EVs will follow the advice.

[4]A pure-strategy NE is a deterministic solution, as opposed to a probabilistic one (e.g., go to charging station c_x, rather than go to c_x with probability 0.5).

The Congestion Game

A congestion game is defined by the 4-tuple

$$\Gamma = (\mathcal{N}, \mathcal{F}, \{\mathcal{S}_i\}, \{\tau_l(n_l), \eta_c(n_c)\}) \ , \tag{6.1}$$

whose elements in our case are as follows.

- The set of players, $\mathcal{N}$, which, as mentioned, correspond to the EVs using the navigation service.
- The set of facilities, $\mathcal{F}$, which is composed of all possible charging stations and road segments included in the road topology, i.e., $\mathcal{F} = \mathcal{C} \cup \mathcal{L}$. Given $\mathcal{F}$, for each player $i \in \mathcal{N}$, a subset $\mathcal{F}_i \subseteq \mathcal{F}$ can be identified, including all facilities that EV i can reach and use on its way to the destination. Clearly, if the road topology is fully connected, then $\mathcal{F}_i = \mathcal{F}$, $\forall i \in \mathcal{N}$.
- Denoting by $\mathcal{P}(\mathcal{F}_i)$ the set of all possible partitions of $\mathcal{F}_i$, $\mathcal{S}_i \subseteq \mathcal{P}(\mathcal{F}_i)$ is the set of viable strategies for EV i, i.e, all groups of facilities that can be used by i. In our context, each strategy in $\mathcal{S}_i$ is composed of:
  (i) one of the charging stations that EV i can reach, along with
  (ii) the road segments forming a route that allows i to go from its current position to the selected charging station, and from there to its final destination.
- For each strategy, the associated utility is the sum of the utilities of each selected facility (either a charging station or a road segment). The utility of a facility is its negated cost. Such a cost is defined as a function mapping the number n_f of players selecting the facility onto a time delay in $\mathbb{R}$. In our context, the cost of a strategy is the sum of 1) the expected waiting time and the service time at the corresponding charging station, and of 2) the travel time on the associated route, from current road segment to destination, via the charging station. We denote the former by $\eta_c(n_c)$, with $c \in \mathcal{C}$ and n_c being the number of players selecting station c. We denote the latter by $\sum_l \tau_l(n_l)$, with the l's being the road segments in the chosen route and n_l the number of players taking segment l.

The game elements are summarized in Tab. 6.1. We stress that, the quantity representing the cost of a charging station or of a road segment, does not depend on the player identity but only on the number of non-player EVs using the facility and on the number of players selecting it. Our definition of the cost therefore complies with the anonymity property of congestion games.

Furthermore, in accordance with the scenario detailed in Section 6.3, we write $\eta_c(n_c)$ so as to account for (a) the number of servers at station c, K_c, (b) the service time, (c)

the number of fully-charged batteries currently available at c, B_c, and (d) the waiting time before an EV can be served. Specifically, we write $\eta_c(n_c)$ as:

$$\eta_c(n_c) = \begin{cases} \sigma & \text{if } w_c < K_c \\ \sigma + \frac{\sigma}{2} & \text{if } K_c \leq w_c < 2K_c \\ \sigma + \frac{\sigma}{2} + \left\lfloor \frac{w_c - K_c}{K_c} \right\rfloor \sigma & \text{if } 2K_c \leq w_c < B_c \\ \rho & \text{if } w_c \geq B_c \end{cases} \tag{6.2}$$

with w_c being the expected number of EVs that the generic player finds at the charging station upon its arrival. Such a value is given by: $w_c = m_c + n_c/2$, where m_c is the number of non-player EVs that the CC estimates to be already at the station upon the arrival of the generic player, and $n_c/2$ is the expected number of other players that have already reached c, if n_c players decide to use such a station. Note that w_c does not account for the precise order of arrival of the single players since the cost cannot depend on the player identity.

In (6.2) the first line corresponds to the case where the generic player finds an idle server, hence its stopping time at c coincides with the time necessary for battery replacement, σ, which is assumed to be constant. The second line, instead, represents the case where all servers are busy but the player finds a server with nobody else waiting to be served (the expected remaining service time of the EV that is currently under service is $\sigma/2$). The third line refers to all servers at c being busy, with EVs already waiting there to be served. Thus, assuming a balanced load, the expression includes the expected time that the generic player has to spend in line. Finally, the last line applies when no more fully-charged batteries are available at the station, and the generic player has to recharge its battery, in a time that is assumed to be constant and equal to ρ.

As mentioned, it has been shown in [105] that congestion games admit at least one pure-strategy that is an NE. However, finding such an equilibrium is, in general, NP-hard [115]. In order to lower the complexity, below we introduce a new game, namely, a game with congestion-averse utilities.

Table 6.1: Comparing congestion games vs. CAGs

	Players	Facilities	Strategies	Strategy Cost
Congestion game	$\mathcal{N}$	$\mathcal{F} = \mathcal{C} \cup \mathcal{L}$	$\forall i \in \mathcal{N} : \mathcal{S}_i = \{\{c_x, l_1, \ldots, l_m\}_x\}$ s.t. $c_x \in \mathcal{C}$ is reachable by i, and $\{l_i\}_{i=1\ldots m} \in \mathcal{L}$ form a route from current i's segment to dest., through c_x	$\eta_c(n_c) + \sum_l \tau_l(n_l)$ (sum over l's $\in$ route)
CAG	$\mathcal{N}$	$\mathcal{F} = \mathcal{C}$	$\forall i \in \mathcal{N} : \mathcal{S}_i = \{\{c_x\}_x, \{c_x, c_y\}_{x,y}, \{c_x, c_y, c_z\}_{x,y,z}\}$ s.t. $c_x, c_y, c_z \in \mathcal{C}$ and reachable by i; $\overline{\mathcal{S}}_i = \{\{c_x\}, \{c_y\}, \{c_z\}\}$	$\eta_c(n_c^{(i)}) + \tau_{i,c}$ (depends on player id)

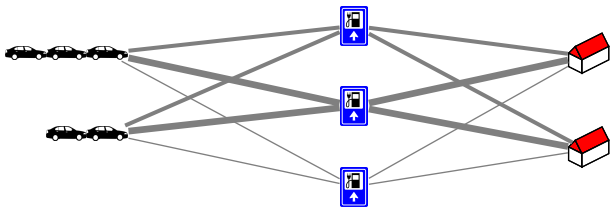


Figure 6.3: Abstract representation of the vehicular scenario where for each EV there is only one possible route toward a given charging station, and from there to its intended destination.

Game Model with Congestion-averse Utilities

Let us now consider the same scenario as above, but assume that, for every EV-charging station pair, there exists only one possible route to take, as depicted in Fig. 6.3. We stress that, although simpler, such a model is still realistic if the CC associates to the EV-charging station pair the route deemed to be the fastest one, based on its recent estimates. Indeed, it is likely that such a route is the most convenient to the EV, hence neglecting the others will not lead to significantly worse performance. This is also confirmed by our results derived in real-world scenarios, shown in Section 6.7.

Under the above assumption, the system can be modeled as a game with congestion-averse utilities (CAG), for which NEs are pure-strategies and can be found in polynomial time [106]. The game is defined as a 4-uple similar to Γ, as in (6.1), however, two main differences exist between CAGs and congestion games:

- in CAGs, it must hold that $\mathcal{S}_i = \mathcal{P}(\mathcal{F}_i)\,, \forall i \in \mathcal{N}$, i.e., all partitions of $\mathcal{F}_i$ are possible strategies, and
- the costs of the facilities can depend on player identities.

The first difference implies that, for each player i, the CC has to consider as viable strategies not a subset but all possible partitions of $\mathcal{F}_i$. A set $\mathcal{F}$ defined as in the case of the congestion game would force the CC to consider non-meaningful strategies where an EV stops at more than one charging station, located either on the same route or on different routes. In order to overcome this issue, as a first step we redefine the set of facilities as $\mathcal{F} = \mathcal{C}$, i.e., we remove the road segments and consider only the charging stations. It follows that the set of facilities that the generic player i can use, $\mathcal{F}_i$, is now given by just the charging stations that the EV can reach. This is a viable choice since, per the initial assumption in this subsection, each EV-charging station pairs is implicitly, and univocally, associated to one route only. As a second step, we prove the lemma below.

Lemma 1. *Consider the game with congestion-averse utilities introduced above, in which each facility has a cost greater than 0. Then, in order to identify a pure-strategy NE, for any player $i \in \mathcal{N}$ it is sufficient to examine the subset of viable strategies $\overline{\mathcal{S}}_i \subseteq \mathcal{S}_i$, such that each strategy in $\overline{\mathcal{S}}_i$ includes one facility only.*

Proof. Players are self-interested and aim at maximizing their utility, i.e., minimize their cost. Recall that costs are positive, thus selecting more than one facility (i.e., charging station) leads to an increased overall cost. A player will therefore always deviate from a strategy profile that lets it use more than one facility. It follows that, in order to find an NE, it is enough to consider as viable strategies for each player the ones that imply the use of one facility only. □

Based on the above result, we can limit our attention to the set of strategies $\overline{\mathcal{S}}_i$, which includes only partitions of $\mathcal{F}_i$ with cardinality equal to 1, and each of them corresponding to only one of the charging stations that EV i can reach.

Next, we leverage the second difference between CAGs and congestion games, i.e., the fact that in CAGs utilities can depend on the player identity. In particular, we define the cost of a charging station c, which can be used by player i, as the total trip time i would incur, and we write it as:

$$\eta_{i,c}(n_c^{(i)}) + \tau_{i,c}\,. \tag{6.3}$$

In (6.3), the first term is the sum of the delay due to the expected waiting time and the charging time at station c, while the second term is the travel time through the route associated to the EV-charging station pair (i, c). Note that both terms depend on the player identity i; furthermore, the following remarks hold.

- $\eta_{i,c}(n_c^{(i)})$ can be obtained from (6.2) by replacing w_c with $m_c^{(i)} + n_c^{(i)}$, and ρ with ρ_i. Indeed, now the CC can account for the number $m_c^{(i)}$ of non-player EVs that it estimates to be at the station upon the arrival of player i. Similarly, $n_c^{(i)}$ is the number of players that the CC estimates to arrive at c before player i does. Finally, the recharging time ρ_i may be different from one player to another, and depend on the remaining battery charge of the EV.

- The travel time $\tau_{i,c}$, associated to the EV-charging station pair (i, c), does not depend on $n_c^{(i)}$, as it now accounts for the vehicular traffic intensity due to all non-player vehicles only (i.e., the contribution of the players is neglected). Indeed, the CAG model cannot track the contribution to the traffic intensity due to players selecting different charging stations but whose route partially overlap. The impact of such an approximation is very limited since typically the number of players is much smaller than the number of all other vehicles traveling over the road topology (see also the results in Section 6.7).

The elements of the CAG are summarized in Tab. 6.1. As mentioned, in the case of CAGs, pure-strategy NEs can be found in polynomial time [106], thus the CC can solve the game with low complexity. In the following, we evaluate the number of strategies that the CC has to process before an NE is found, as well as the social utility corresponding to such an NE, i.e., how good the NE is from the system performance viewpoint. We

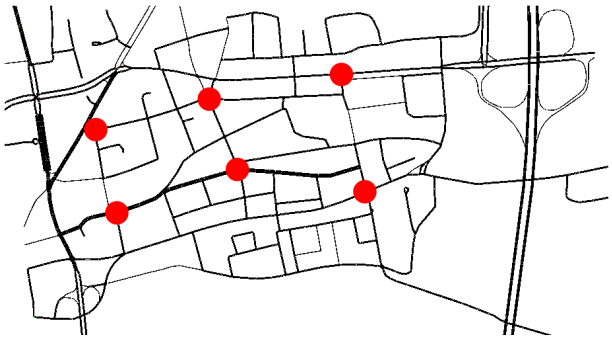


Figure 6.4: Road topology: red dots highlight the six charging stations.

also show that, in spite of its low complexity, the CAG model approximates very well the previous (most general) scenario where multiple routes may exist for any EV-charging station pair.

6.6 Simulation Scenario

In order to show the benefits that can be obtained through our game-theoretic approach, we use a real-world road topology representing a 3×2 km^2 section of the urban area of Ingolstadt, Germany [116], depicted in Fig. 6.4. The vehicle mobility has been synthetically generated using the SUMO simulator [87], with a time granularity of 0.1 s. The mobility trace is representative of 30-minute-long road traffic and of average traffic intensity in the area. We stress that we preferred a synthetic trace over real-world ones, e.g., taxi or bus traces, since these only include a small portion of the car traffic and the represented vehicles have predetermined routes from which they cannot deviate. The number of vehicles simultaneously present in our road topology is a varying parameter of the system, and the average vehicle trip time clearly depends on such a value.

The scenario includes 6 charging stations, which are placed on the main arteries of the road topology, as portrayed by the red dots in Fig. 6.4. The number of servers at each station may vary; namely, two stations have 2 servers, other two have 6 servers and the remaining ones have 4 and 10 servers each. We assume that fully-charged batteries are always available at the charging stations, thus the service time is considered to be constant and equal to 3 minutes.

Without loss of generality, all vehicles are assumed to be electric. The average number of EVs that resort to the navigation service is a varying parameter in our simulations. The time instant at which an EV enters the low-battery status and asks the CC for advice is uniformly distributed over its trip time, i.e., the time interval since the EV enters the road topology till it leaves. For simplicity, we neglect the presence of EVs with low battery that do not use the navigation service. Also, unless otherwise specified, we assume that all EV drivers are rational.

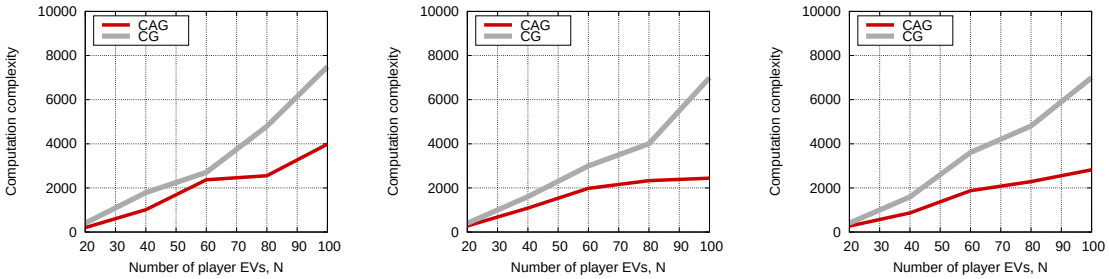


Figure 6.5: Average computational complexity vs. number of players, when they are 20% (left), 40% (middle), 60% (right) of all vehicles. CAG and congestion game (CG) are compared.

The navigation service is provided via the cellular network, through which an EV may issue a query to and receive a response from the CC without significant delay. However, alternative solutions exploiting 802.11p-based roadside units could be considered as well. As for the CC, we consider that information on the number of EVs currently waiting at a charging station to be served, as well as on the traffic conditions, is acquired and processed every 10 seconds. The requests for the navigation service sent by the EVs are instead processed by the CC every $T = 60$ s.

6.7 Results

We now show the performance that is attained through our approach, and compare it to the results obtained when a centralized optimization problem is solved at the CC as well as when a greedy selection of charging station and route is adopted.

In order to derive the results in the cases where the CC generates its advice from the solution of the CAG or of the congestion game, we proceed as follows. Every time interval T, the CC solves the game considering as players the EVs from which it has received a request. To do so, the CC starts from a random strategy profile, i.e., a random assignment of the facilities to the players. In the case of the congestion game, it assigns both the charging station and the corresponding route, while in the CAG, it assigns only the charging station and associates to each player-charging station pair the fastest route that takes the EV from its current road segment to the station, and from there to its destination. Player payoffs (i.e., trip times) are then computed through SUMO in the scenario described in Section 6.6. To derive the trip times, we assume that every non-player vehicle takes its originally-planned route, while players will conform to the CC's advice, hence they will follow the suggested route.

Given the current strategy profile and player payoffs, the CC examines other strategies according to the solution algorithm in [106] for the CAG, and to the one in [117, Ch.7] for the congestion game. For every strategy, player payoffs are computed via SUMO as before. If a more convenient strategy is found for any of the players, then the new strategy

is adopted and the whole procedure is repeated until an NE is reached. Unless otherwise specified, we consider that the CC takes the first NE it finds as the solution of the game.

For both the CAG and the congestion game, we evaluate the computational complexity, i.e., the number of strategies that the CC has to examine before reaching the game solution, which also corresponds to the number of SUMO runs. Then, we calculate the per-player trip time associated to such a solution. All results are averaged over 10 runs. We compare such values with the trip time obtained through the techniques described below.

Optimal: the solution that the CC can obtain by minimizing the trip time averaged over all EVs that ask for advice. This solution in general is not an NE, thus it may not be followed by rational drivers.

Greedy: the CC only disseminates information on the roads travel time, and on the occupancy and the charging time at the stations. Based on this knowledge, each EV independently makes its own decision by selecting the charging station and the route that are deemed to minimize its own trip time. Note that, in this case, the CC just informs the EVs without providing any advice, and the EV decision is taken disregarding the presence of other vehicles looking for a charging station.

Fig. 6.5 depicts the number of strategies that the CC has to examine before the solution to the game is found, for both the CAG and the congestion game (CG). We stress that the CC returns its advice to EV drivers only once the game solution (which is a pure-strategy NE) has been reached, thus the computational burden is solely carried by the CC. The three plots in the figure refer to the cases where the average number of EVs that are low in battery and ask for advice (i.e., players) is, respectively, 20%, 40% and 60% of the average total number of vehicles simultaneously present in the road topology. Thus, for a given average number of players, the plots (from left to right) correspond to decreasing values of the average total number of vehicles, i.e., a decreasing number of non-player EVs. As an example, for an average number of players equal to 60, the left plot corresponds to an average total number of EVs equal to 300, the middle plot to 150, and the right one to 100. Although 60% may seem an unreasonably high percentage, it was chosen to stress the system.

As expected, the complexity of the congestion game is always higher than that of the CAG and, in both cases, it increases as the number of players grows. In particular, for our range of player numbers, the CC always examines less than 4000 strategies before finding the solution in the case of the CAG, and less than 8000 in the case of the congestion game. We remark that one SUMO run only takes few seconds, hence the simulation impact is very limited.

The plots also provide a striking comparison between the CAG and the congestion game. While the complexity of the former remains remarkably low, the complexity of the latter increases severely as the number of players grows beyond 60. On the contrary, the total number of EVs in the road topology has just a marginal impact on both the CAG

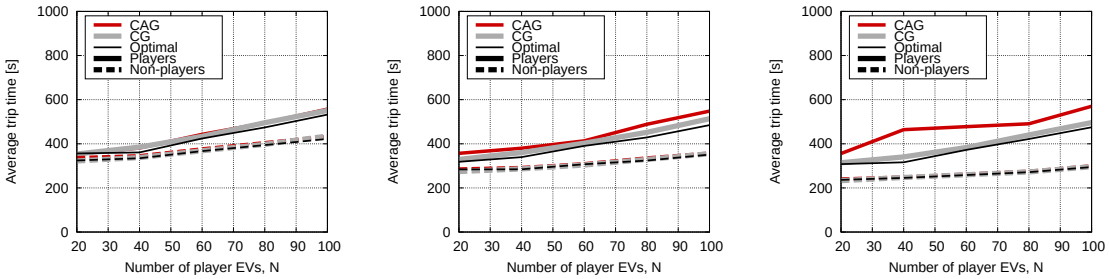


Figure 6.6: Average per-EV trip time as a function of the number of players, when they represent 20% (left), 40% (middle), 60% (right) of all vehicles. CAG and congestion game (CG) are compared against the optimal, for both player and non-player vehicles.

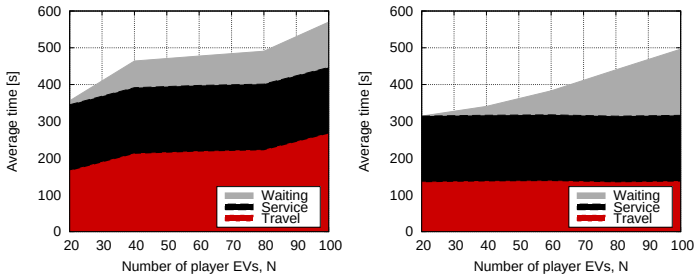


Figure 6.7: Trip time breakdown for the CAG (left) and congestion game (right), when players are 60% of the total number of vehicles.

and the congestion game solution time. These results indicate that the CAG model is highly scalable, hence it can be successfully applied even to very large, crowded system scenarios.

Next, one may wonder whether the solution obtained through the CAG is as good as the one of the congestion game, or if the gain in complexity we have with the CAG takes a high toll in terms of system performance. To answer this question, in Fig. 6.6 we show the average vehicle trip time, for both player and non-player EVs, again as the number

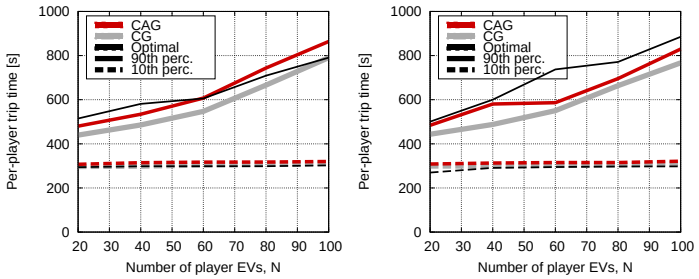


Figure 6.8: CAG, congestion game and optimal: 10th and 90th percentile of the per-player trip time, vs. number of players, when they are 20% (left) and 60% (right) of all vehicles.

of players is 20%, 40% and 60% of the total number of vehicles. The performance corresponding to the solutions of the two games are also compared to that of the centralized optimal solution.

The figure shows that the average trip times of player and non-player EVs have the same qualitative behavior, with the former clearly being higher than the latter since players stop at a charging station during their trip. Also, comparing the three plots, it can be seen that the smaller the total number of vehicles simultaneously present in the road topology, the lower the traffic intensity and the shorter the average per-EV trip time. As for the comparison among the CAG, the congestion game and the optimal, the difference in performance can be barely noticed when the players are 20% and 40% of the total number of EVs (left and middle plots of Fig. 6.6). When the percentage of players is large (right plot), the difference with respect to the optimal is limited in the case of the CAG, and it is again unnoticeable for the congestion game. This indicates that neglecting the contribution of player EVs to the travel time makes the CAG model less precise only when players represent the majority of vehicles on the road topology.

Fig. 6.7 confirms such an observation. The figure highlights the different contributions to the average per-player trip time, due to the waiting time at the charging station, the service time (which is constant) and the travel time. The results refer to the CAG (left plot) and to the congestion game (right plot), when the players are 60% of all vehicles. It can be seen that the difference between the two game models mainly resides in the travel time, which is higher when the CAG solution is adopted.

Fig. 6.8 depicts the 10th (dashed line) and the 90th (solid line) percentiles of the per-player trip time, when players are 20% (left) and 60% (right) of all vehicles. In the case of the 10th percentile, the difference, among the solution of the CAG, that of the congestion game and the optimal, can be barely detected. As for the 90th percentile, it can be observed that, when the optimal solution is adopted, a fraction of player EVs may experience a significantly longer trip time than under the congestion game or the CAG. This suggests that applying the optimal solution may lead to higher unfairness in the user

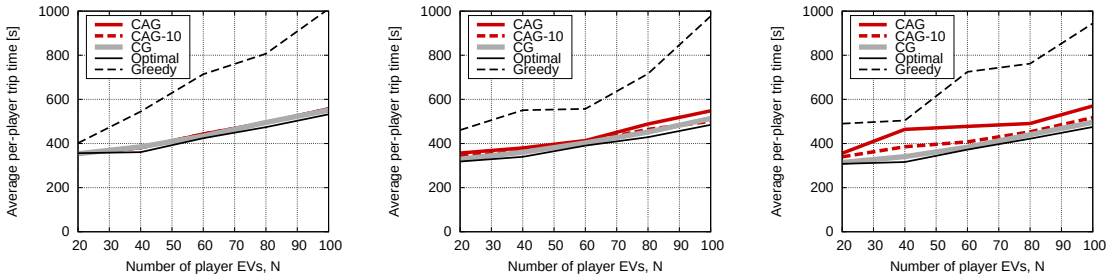


Figure 6.9: Average per-player trip time vs. number of players, when they represent 20% (left), 40% (middle), 60% (right) of all vehicles. Comparison among CAG, congestion game, optimal, and greedy. CAG-10 refers to the case where the CC takes as a solution of the game the best among the first 10 NEs it finds.

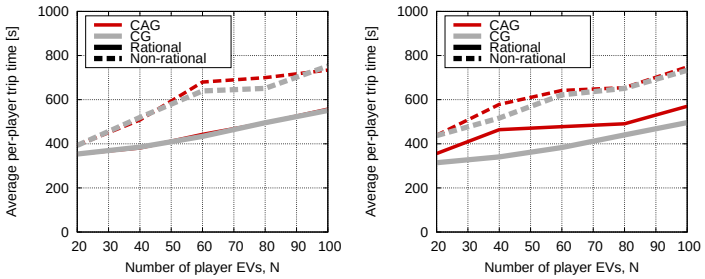


Figure 6.10: Average per-player trip time vs. number of players, when they are 20% (left) and 60% (right) of all vehicles. Comparison between rational and non-rational users.

performance.

We now investigate the benefit of our approach with respect to the aforementioned greedy scheme. Recall that the greedy technique assumes the EVs to have periodically updated information about road traffic and status of the charging stations. In spite of this, Fig. 6.9 clearly shows that a greedy approach cannot cope with the other techniques in terms of performance: the degradation that is observed is indeed severe and becomes exceedingly high as the number of players increases. Intuitively, this is due to many users selecting the (currently) least crowded station, which suddenly becomes overloaded (as in the well-known route-flapping effect). Fig. 6.9 also depicts the performance of the CAG when the CC does not solve the game using the first NE that is reached, but the NE that minimizes the average per-player trip time among the first 10 it finds. In the plots, we label this curve by CAG-10. Interestingly, such a simple enhancement to the solution procedure makes the CAG approach as effective as the congestion game and the optimal, without impairing its scalability (see in particular the right plot).

In conclusion, not only modeling the system through a CAG is a feasible, practical approach to the problem, but its solution also leads to a performance that is remarkably close to the optimum and much better than that attained with a greedy scheme.

At last, we consider the case where not all EVs that resorted to the navigation service are willing to follow the advice of the CC. We call such EVs non-rational users, and assume that they will act according to the greedy scheme. The results portrayed in Fig. 6.10 refer to the case where there is an equal number of rational and non-rational users. They show that non-rational users, on average, experience a higher trip time than rational ones. Such a difference in performance is particularly evident, as the number of players increases. This further confirms that our game-theoretic approach always leads to solutions (i.e., advice from CC) that are convenient to the EVs thus increasing the user satisfaction.

6.8 Conclusions

Leveraging the use of ITS, we envisioned the availability of a navigation service that provides electric vehicles (EVs) that are low in battery with advice on the charging station to use and the route to take. We focused on how to determine such advice so that rational EV drivers find it convenient to themselves and they are willing to follow it.

After showing that traditional optimization approaches fail to achieve the above goal, we considered a realistic scenario and modeled the problem as a congestion game, for which at least one pure-strategy Nash equilibrium exists (i.e., a solution that all EVs find it satisfactory). Then, in order to lower the complexity, we introduced a game with congestion-averse utilities (CAG) that applies to a slightly simpler scenario but for which an NE can be found in polynomial time. We assessed the performance of our approach through SUMO and under a real-world vehicular environment. The results show that using CAGs, not only is a viable, scalable technique, but it also leads to an average per-EV trip time that is remarkably close the minimum that can be found through a traditional optimization approach.

Future work will consider other road topologies as well as vehicular traffic scenarios, and it will address the case where the information collected through ITS and available at the CC may be partial or not fully accurate.

Chapter 7

Conclusions and Future Work

In this thesis, different problems have been investigated. For most of them, we came out with easy, original and effective solutions. In the following, you can find the most valuable results.

The first studied problem was how to maintain connectivity among RSUs and moving vehicles in case of UDP-based multimedia streams. We considered vehicles (e.g., cars, buses or streetcars) that connect to different roadside mesh nodes as they move in urban environment. We assessed the performance of different routing protocols, both in simulation and in the real field, but we realized that no one fit our needs. Thus, we designed a new routing protocol able to support vehicular mobility called sw-BATMAN. We implemented it in our vehicular testbeds, along with a channel selection mechanism and a seamless handover procedure. We proved the feasibility of our solution and we opened interesting perspectives in the use of mesh networks for the support of UDP-based service to vehicular users.

Studying the first problem, we became conscious that the 5-GHz bands offer limited capacity channels, in comparison to the broad range of services that are envisioned in vehicular networks. We therefore explored the benefit of using UHF bands for the transmission of control messages, so as to acquire more capacity. Specifically, we focused on content downloading, and design a protocol that leverages the UHF band for control messages (aimed at locating vehicles and collecting requests) and the high-throughput, 5-GHz bands for data delivery. We assessed the benefits of exploiting UHF bands, providing much larger coverage than the 5-GHz frequencies, through a vehicular testbed. We proved that our solution introduces a 3x throughput gain in content delivery with respect to the case where only 5-GHz bands are used.

Then, we studied where the RSUs have to be installed to provide high users coverage for Content Dissemination and Downloading. We reviewed previous work that has dealt with such an issue, and we presented a new strategy called MCTTP (Maximum Coverage with Time Threshold Problem) aimed to guarantee that a large number of vehicles travel under the coverage of one or more RSUs for a sufficient amount of time.

We clearly demonstrated that RSU deployment cannot be addressed through random or intuitive placements, since neither one represents, in general, an optimal solution to the RSU deployment problem.

Since urban traffic jams are an actual problem, we thought that we could use the deployed infrastructure to provide navigation hints to the drivers, trying to prevent the congestion problem. Thus, we invented a method to optimize the urban traffic layout using basic heuristics and computationally efficient simulations. Instead of modeling an entire urban map with hundreds of crossroads, we simulated each typology of intersection to understand how it responds to different traffic patterns and intensities. Then, this knowledge is leveraged to allow the computation of minimal delay route on the complete road map. We proved that, if the drivers follow the information provided by the navigation system, our strategy prevents traffic jam and maintain the average travel time close to the optimum.

Another application to exploit the infrastructure, in an environmental friendly context, is by assisting the Electric Vehicle (EV) drivers that need to recharge or to substitute their batteries. Drivers of EVs that are low in battery may ask a navigation service for advice on which charging station to use and route to take. A rational driver will follow the received advice, provided there is no alternative choice that lets the driver reach its destination in a shorter time. Therefore, we represented this scenario using a game-theoretic model and we assessed its performance through simulations under a real-world vehicular environment. We showed that the average per-EV trip time yielded by our model is very close to the one attained by solving a centralized optimization problem that minimizes such quantity. This is an important result, as minimizing the average per-EV trip time implies reduced road traffic congestion and energy consumption, as well as higher user satisfaction.

Future work will focus on improving the previous studies, refining the presented solutions and collecting more measurements in complex topologies. Moreover, additional effort will be invested in designing new applications for Vehicular Networks with Infrastructure, envisioning also cooperation among vehicles exploiting the well-known Vehicle-to-Vehicle communication paradigm.

Bibliography

[1] A. Capone, S. Napoli, A. Pollastro, "MobiMESH: An Experimental Platform for Wireless MESH Networks with Mobility Support," *ACM QShine, Workshop on Wireless mesh: moving towards applications (WiMESHNets)*, Waterloo, Canada, Aug. 2006.

[2] Y. Amir, C. Danilov, M. Hilsdale, R. Musuloiu-Elefteri, N. Rivera, "Fast Handoff for Seamless Wireless Mesh Networks," *ACM MobiSys*, Uppsala, Sweden, June 2006, pp. 83–95.

[3] I. Ramani, S. Savage, "SyncScan: Practical Fast Handoff for 802.11 Infrastructure Networks," *IEEE INFOCOM*, Miami, FL, Mar. 2005, pp. 675–684.

[4] V. Brik, A. Mishra, S. Banerjee, "Eliminating Handoff Latencies in 802.11 WLANs Using Multiple Radios: Applications, Experience, and Evaluation," *ACM SIGCOMM*, Berkeley, CA, Oct. 2005, pp. 27–37.

[5] R. Draves, J. Padhye, B. Zill, "Routing in Multi-radio, Multi-hop Wireless Mesh Networks," *ACM MobiCom*, New York, NY, 2004, pp. 114-128.

[6] S. Liese, D. Wu, P. Mohapatra, "Experimental Characterization of an 802.11b Wireless Mesh Network," *International Conference on Wireless Communications and Mobile Computing (IWCMC)*, Vancouver, Canada, July 2006, pp. 587–592.

[7] J. Robinson, K. Papagiannaki, C. Diot, X. Guo, L. Krishnamurthy, "Experimenting with a Multi-Radio Mesh Networking Testbed," *International Workshop on Wireless Network Measurement (WinMee)*, Riva del Garda, Trentino, Apr. 2005.

[8] IEEE P802.11p /D11.0, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment : Wireless Access in Vehicular Environments, Mar. 2010.

[9] IEEE P802.11/D2.02, Draft Standard, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, Amendment 10: Mesh Networking, Sept. 2008.

[10] IETF Mobile Ad-hoc Networks (manet) Working Group, `http://datatracker.ietf.org/wg/manet/charter/`.

[11] A. Neumann, C. Aichele, M. Lindner, S. Wunderlich, "Better Approach To Mobile Ad-hoc Networking (B.A.T.M.A.N.)," draft-wunderlich-openmesh-manet-routing-00, *IETF Work In Progress Internet-Draft*, Apr. 2008.

[12] C. Perkins, E. Belding-Royer, S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," *IETF Experimental RFC 3561,* July 2003.

[13] T. Clausen, P. Jacquet, "Optimized Link State Routing Protocol (OLSR)," *IETF Experimental RFC 3626,* Oct. 2003.

[14] B. N. Karp, H. T. Kung "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," *ACM/IEEE MobiCom,* Boston, MA, Aug. 2000, pag. 243–254.

[15] J. Ott, D. Kutscher, "Drive-thru Internet: IEEE 802.11b for Automobile Users," *IEEE INFOCOM,* Hong Kong, Mar. 2004, pag. 362–373.

[16] M. Wellens, B. Westphal, P. Mahonen, "Performance Evaluation of IEEE 802.11-based WLANs in Vehicular Scenarios," *IEEE VTC,* Dublin, Ireland, Apr. 2007, pp. 1167–1171.

[17] M. Jerbi, S. M. Senouci, "Characterizing Multi-hop Communication in Vehicular Networks," *IEEE WCNC,* Las Vegas, Nevada, Apr. 2008, pp. 3309–3313.

[18] G. Korkmaz, E. Ekici, F. Ozguner, "Supporting Real-time Traffic in Multihop Vehicle-to-infrastructure Networks," *Transportation Research, Part C (Elsevier),* Vol. 18, No. 3, June 2010, pp. 376–392.

[19] A. Böhm, M. Jonsson, "Handover in IEEE 802.11p-based Delay-Sensitive Vehicle-to-Infrastructure Communication," *Technical Report IDE0924,* Halmstad University, Sweden, 2007.

[20] A. Böhm, M. Jonsson, "Position-based Data Traffic Prioritization in Safety-critical Vehicle-to-infrastructure Communication", *IEEE ICC Vehicular Networks Applications Workshop,* Dresden, Germany, June 2009.

[21] A. Böhm, M. Jonsson, "Supporting Real-time Data Traffic in Safety-critical, Vehicle-to-infrastructure Communication," *IEEE LCN Workshop on User Mobility and Vehicular Networks* , Montreal, Canada, Oct. 2008, pp. 614–621.

[22] J. Miller, "Vehicle-to-Vehicle-to-Infrastructure (V2V2I) Intelligent Transportation System Architecture," *IEEE Intelligent Vehicles Symposium,* Eindhoven, Netherlands, June 2008, pp. 715–720.

[23] IETF Network Mobility (NEMO) Working Group Charter, `http://www.ietf.org/html.charters/nemo-charter.html`.

[24] V. Devarapalli, R. Wakikawa, A. Petrescu, P. Thubert, "Network Mobility (NEMO) Basic Support Protocol," *IETF, Request For Comments 3963,* Jan. 2005.

[25] V. Angelakis, M. Genetzakis, N. Kossifidis, K. Mathioudakis, M. Ntelakis, S. Papadakis, N. Petroulakis, V. A. Siris, "Heraklion MESH: An Experimental Metropolitan Multi-Radio Mesh Network," *ACM WiNTECH,* Montreal, Canada, Sept. 2007, pp. 93–94.

[26] Y. Su, T. Gross, "Validation of a Miniaturized Wireless Mesh Network Testbed," *ACM WiNTECH,* San Francisco, CA, Sept. 2008, pp. 25-32.

[27] H. Aache, V. Conan, L. Lebrun, S. Rousseau, "A Load Dependent Metric for Balancing Internet Traffic in Wireless Mesh Networks," *IEEE MASS, Mesh-Tech*, Atlanta, GA, Oct. 2008, pp. 629–634.

[28] D. M. Shila, T. Anjali, "Load Aware Traffic Engineering for Mesh Networks," *Computer Communications*, Vol. 31, No. 7, May 2008, pp. 1460–1469.

[29] J. Wang, Y. Yang, R. Kravets, "Designing Routing Metrics for Mesh Networks," *IEEE WiMesh*, Santa Clara, CA, Sept. 2005.

[30] A. Neishaboori, G. Kesidis, "SINR-Sensitive Routing in Wireless 802.11 Mesh Networks," *IEEE MASS, Mesh-Tech*, Atlanta, GA, Oct. 2008, pp. 623–628.

[31] R. Baumann, S. Heimlicher, V. Lenders, M. May, "Routing Packets into Wireless Mesh Networks," *IEEE WiMob*, New York City, NY, Oct. 2007, pp. 38–38.

[32] S. Annese, C. Casetti, C.-F. Chiasserini, P. Cipollone, A. Ghittino, M. Reineri, "Assessing Mobility Support in Mesh Networks," *ACM WiNTECH*, Beijing, China, Sept. 2009, pp. 19–26.

[33] X. Zhang, H. Su, "Cluster-Based Multi-Channel Communications Protocols in Vehicle Ad Hoc Networks," *IEEE Wireless Communications Magazine*, Vol. 13, No. 5, Oct. 2006, pp. 44–51.

[34] M. Raya, P. Papadimitratos, J.-P. Hubaux, "Securing Vehicular Communications," Vol. 13, No. 5, *IEEE Wireless Communications Magazine*, Oct. 2006, pp. 8–15.

[35] CAR 2 CAR Communication Consortium Manifesto, `http://www.car-to-car.org/fileadmin/downloads/C2C-CC_manifesto_v1.1.pdf`, Aug. 2007.

[36] ETSI WG4, Intelligent Transport Systems (ITS) - European profile standard for the physical and medium access control layer of Intelligent Transport Systems operating in the 5 GHz frequency band, Jan. 2010.

[37] V. Rai, F. Bai, J. Kenney, K. Laberteaux, "Cross-Channel Interference Test Results: A report from the VSC-A project," IEEE 802.11 11-07-2133-00-000p, July 2007.

[38] M. Marina, S. Das, "On-demand Multipath Distance Vector Routing in Ad Hoc Networks," *International Conference for Network Protocols (ICNP)*, Riverside, CA, Nov. 2001, pp. 14-23.

[39] Z. Ye, T. S. Krishnamurthy, "A Framework for Reliable Routing in Mobile Ad Hoc Networks," *IEEE INFOCOM*, San Francisco CA, Apr. 2003, pp. 270-280.

[40] D. Johnson, G. Hancke, "Comparison of Two Routing Metrics in OLSR on a Grid Based Mesh Network," *Ad Hoc Networks*, Vol. 7, No. 2 Mar. 2009, pp. 374–387.

[41] S. Annese, C. Casetti, C.-F. Chiasserini, N. Di Maio, A. Ghittino, M. Reineri, "A Vehicular Network with Infrastructure in a Woodland Area: Testbed Results," *Technical Report*, Politecnico di Torino, May 2010, `http://www.telematica.polito.it/casetti/TechRep_VGTestbed_May10.pdf`.

[42] `http://www.youtube.com/watch?v=FDzSkNXEsZg`

[43] G. Staple, K. Werbach, "The End of Spectrum Scarcity [Spectrum Allocation and Utilization]," *IEEE Spectrum*, vol. 41, no. 3, pp. 48-52, 2004.

[44] D. Niyato, E. Hossain, P. Wang, "Optimal Channel Access Management with QoS Support for Cognitive Vehicular Networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 5, pp. 573-591, 2011.

[45] K. Fawaz, A. Ghandour, M. Olleik, H. Artail, "Improving Reliability of Safety Applications in Vehicle Ad hoc Networks through the Implementation of a Cognitive Network," *IEEE International Conference on Telecommunications*, Doha, Qatar, Apr. 2010.

[46] N. Kirsch, B.M. OConnor, "Improving the Performance of Vehicular Networks in High Traffic Density Conditions with Cognitive Radios," *IEEE Intelligent Vehicles Symposium*, Baden-Baden, Germany, June 2011.

[47] W. Kim, S.Y. Oh, M. Gerla, K.C. Lee, "CoRoute: A New Cognitive Anypath Vehicular Routing Protocol," *IEEE International Wireless Communications and Mobile Computing Conference (IWCMC)*, Istanbul, Turkey, July 2011.

[48] R. Sevlian, C. Chun, I. Tan, A. Bahai, K. Laberteaux, "Channel Characterization for 700 MHz DSRC Vehicular Communication," *Journal of Electrical and Computer Engineering*, Jan. 2010.

[49] G. Marfia, M. Roccetti, A. Amoroso, M. Gerla, G. Pau, J.-H. Lim, "Cognitive Cars: Constructing a Cognitive Playground for VANET Research Testbeds," *ACM Conference on Cognitive Radio and Advanced Spectrum Management (CogART 2011)*, Barcelona, Spain, Oct. 2011.

[50] A. Masini, G. Mazzini, A. Ghittino, M. Maglioli, N. Di Maio, G. Riva, "WhiteFi: the Usage of UHF Frequencies for Bi-directional Services in Mountainous Scenarios," *International Conference on Electromagnetics in Advanced Applications*, Turin, Italy, Sep. 2011.

[51] G. Marfia, G. Pau, E. Giordano, E. De Sena, M. Gerla, "Evaluating vehicle network strategies for downtown Portland: opportunistic infrastructure and importance of realistic mobility models,". *ACM MobiOpp 2007*, San Juan, Puerto Rico, 2007

[52] V. Bychkovsky, B. Hull, A.K. Miu, H. Balakrishnan, S. Madden, "A Measurement Study of Vehicular Internet Access Using In Situ Wi-Fi Networks," *ACM/IEEE MobiCom 2006*, Los Angeles, CA, USA

[53] M. Fiore, J.M. Barcelo-Ordinas, "Cooperative download in urban vehicular networks," *IEEE MASS 2009*, Macau, China

[54] O. Trullols-Cruces, M. Fiore, C. Casetti, C.-F. Chiasserini, J.M. Barcelo-Ordinas, "Planning roadside infrastructure for information dissemination in intelligent transportation systems," *Elsevier Computer Communications*, vol. 33, no. 4, 2010

[55] Y. Ding, C. Wang, L. Xiao, "A Static-Node Assisted Adaptive Routing Protocol in Vehicular Networks," *ACM VANET Workshop 2007*, Montreal, Canada, 2007

[56] C. Lochert, B. Scheuermann, C. Wewetzer, A. Luebke, M. Mauve, "Data aggregation and roadside unit placement for a VANET traffic information system," *ACM VANET Workshp 2008*, San Francisco, CA, USA, 2008

[57] J. Ahn, B. Krishnamachari, F. Bai, L. Zhang, "Optimizing Content Dissemination in Heterogeneous Vehicular Networks," *Technical Report*, 2011

[58] Z. Zheng, P. Sinha, S. Kumar, "Alpha Coverage: Bounding the Interconnection Gap for Vehicular Internet Access," *IEEE INFOCOM 2009*, Rio de Janeiro, Brasil,

2009

[59] Z. Zheng, Z. Lu, P. Sinha, S. Kumar, "Maximizing the Contact Opportunity for Vehicular Internet Access," *IEEE INFOCOM 2010,* San Diego, CA, USA, 2010

[60] F. Malandrino, C. Casetti, C.-F. Chiasserini, M. Fiore, "Content Downloading in Vehicular Networks: What Really Matters," *IEEE INFOCOM 2011,* Shanghai, China, 2011

[61] A. Abdrabou W. Zhuang, "Probabilistic Delay Control and Road Side Unit Placement for Vehicular Ad Hoc Networks with Disrupted Connectivity," *IEEE Journal on Selected Areas in Communications,* vol. 29 no. 1, 2011

[62] J. Zhao, G. Cao, "VADD: Vehicle-Assisted Data Delivery in Vehicular Ad Hoc Networks," *IEEE Transactions on Vehicular Technology,* vol. 57 no. 3, 2008

[63] S. Yoon, H.Q. Ngo, C. Qiao, "On Shooting a Moving Vehicle with Data Flows," *IEEE MOVE Workshop 2007,* Anchorage, AK, USA, 2007

[64] J. Whitbeck, Y. Lopez, J. Leguay, V. Conan, M. Dias de Amorim, "Relieving the Wireless Infrastructure: When Opportunistic Networks Meet Guaranteed Delays," *IEEE WoWMoM 2011,* Lucca, Italy, 2011

[65] S.O. Krumke, M.V. Marathe, D. Poensgen, S.S. Ravi, H.-C. Wirth, "Budgeted maximum graph coverage. Lecture Notes In Computer Science," 2002

[66] R.M. Whitaker, L. Raisanen, S. Hurley, "The infrastructure efficiency of cellular wireless networks," *Elsevier Computer Networks,* vol. 48 no. 6, 2005

[67] H. Kim, Y. Seok, N. Choi, Y. Choi, T. Kwon, "Optimal multi-sink positioning and energy-efficient routing in wireless sensor networks," *ICOIN 2005,* Jeju Island, Korea, 2005

[68] E.I. Oyman, C. Ersoy, "Multiple sink network design problem in large scale wireless networks," *IEEE ICC,* Paris, France, 2004

[69] W.Y. Poe, J.B. Schmitt, "Minimizing the Maximum Delay in Wireless Sensor Networks by Intelligent Sink Placement," *Technical Report 362/07,* University of Kaiserslautern, Germany, 2007

[70] L. Qiu, R. Chandra, K. Jain, M. Mahdian, "Optimizing the placement of integration points in multi-hop wireless networks," *IEEE ICNP 2004,* Berlin, Germany, 2004

[71] Y. Hu, Y. Xue, Q. Li, F. Liu, G.Y. Keung, B. Li, "The sink node placement and performance implication in mobile sensor networks," *Journal on Mobile Networks and Applications,* no. 14, 2009

[72] A.A. Ageev, M.I. Sviridenko, "Approximation algorithms for maximum coverage and max cut with given sizes of parts," *Lecture Notes in Computer Science,* 1999

[73] D. Pisinger, "Where are the hard knapsack problems?," *Computers and Operations Research 32,* 2004

[74] V. Naumov, "Realistic vehicular traces," from http://lst.inf.ethz.ch/ad-hoc/car-traces, April 6, 2011,

[75] N. Cetin, A. Burri, K. Nagel, "A Large-scale Multi-agent Traffic Microsimulation Based on Queue Model," *STRC 2003,* Ascona, Switzerland, 2003

[76] J. Karedal, N. Czink, A. Paier, F. Tufvesson, A.F. Molisch, "Path loss modeling for vehicle-to-vehicle communications," *IEEE Transactions on Vehicular Technology*, 2011

[77] F. Aidouni, M. Latapy, C. Magnien, "Ten weeks in the life of an eDonkey server," *Hot-P2P 2009*, Rome, Italy, 2009

[78] D.N. Cottingham, J.J. Davies, A.R. Beresford, "Congestion-Aware Vehicular Traffic Routing Using WiFi Hotspots," *Technical Report*, University of Cambridge, 2009

[79] Forest Service Northern Region Engineering, "Cost Estimating Guide for Road Construction," *Technical Report*, United States Department of Agriculture, February 2009

[80] E. Brockfeld, R. Barlovic, A. Schadschneider, M. Schreckenberg, "Optimizing Traffic Lights in a Cellular Automaton Model for City Traffic," *Physical Review E Journal*, October 2001

[81] W. Kim, M. Gerla, "NAVOPT: Navigator Assisted Vehicular route OPTimizer," *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2011 Fifth International Conference on*, Seoul, Korea, July 2011

[82] L. Fratta, M. Gerla, and L. Kleinrock, "The flow deviation method: An approach to store-and-forward communication network design," *Networks, an Internation jounal*, vol. 3, no. 2, January 1973

[83] Y. Ohara, M. Bhatia, N. Osamu, J. Murai, "Route Flapping Effects on OSPF," *Applications and the Internet Workshops, IEEE/IPSJ International Symposium on, p. 232, 2003 Symposium on Applications and the Internet Workshops (SAINT 2003 Workshops)*, 2003

[84] G.F. Newell, "A simplified car-following theory: a lower order model," *Elsevier, Transportation Research Part B: Methodological,*, vol. 36, no. 3, March 2002

[85] Transport and Road Research Laboratory, R.M. Kimber, E.M. Hollis, "Traffic Queues and Delays at Road Junctions," *Transport and Road Research Laboratory*, 1979

[86] D. Husch, J. Albeck, "Intersection Capacity Utilization: Evaluation Procedures for Intersections and Interchanges," *Trafficware*, 2003 Edition

[87] SUMO, Simulation of Urban Mobility, `http://sumo.sourceforge.net/`.

[88] S. Krauss, "Microscopic Modeling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics," *Hauptabteilung Mobilitt und Systemtechnik des DLR Kln*, Universitat zu Koln, 1998.

[89] M. Fiore, "Mobility Models in Inter-Vehicle Communications Literature," *Technical Report*, Politecnico di Torino, 2006

[90] I. Prigogine, R. Herman, "A Two-Fluid Approach to Town Traffic, " *Science*, vol. 204, no. 4389, April 1979

[91] E. Gustafsson, R. Ronngren, "Fluid traffic modelling in simulation of a call admission control scheme for ATM networks," *MASCOTS '97., Proceedings Fifth*

International Symposium on, Haifa, Israel, January 1997

[92] D.C. Gazis, R. Herman, R.W. Rothery, "Nonlinear Follow-The-Leader Models of Traffic Flow," *Operations Research*, vol. 9, no. 4, August 1961

[93] K. Nagel, M. Schreckenberg, "A cellular automaton model for freeway traffic," *Journal de Physique I*, December 1992

[94] O. Biham, A.A. Middleton, D. Levine, "Self-organization and a dynamical transition in traffic-flow models," *Physical Review A Journal*, vol. 46, no. 10, November 1992

[95] S. Scellato, L. Fortuna, M. Frasca, J. Gmez-Gardees, V. Latora, "Traffic optimization in transport networks based on local routing," *The European Physical Journal B*, vol. 73, no. 2, December 2009

[96] B.K. Mohandas, R. Liscano, O.W.W. Yang, "Vehicle Traffic Congestion Management in Vehicular ad-hoc networks," *Local Computer Networks, 2009. LCN 2009. IEEE 34th Conference on*, Zrich, Switzerland, October 2009

[97] H. Ceylan, M.G.H. Bell, "Traffic signal timing optmization based on genetic algorithm approach including drivers routing," *Transportation Research Part B 38*, 2004

[98] Y. Yu, B. Liu, T. Wu, M. Liu, "Flow-based Travel Plan via VANET," *International Journal of Digital Content Technology and its Applications*, vol. 5, no. 6, June 2011.

[99] A. Pascale, M. Nicoli, "Adaptive Bayesian network for traffic flow forecasting," *IEEE SSP 2011*, Nice, France, June 2011

[100] C. Osorio, M. Bierlaire, "A surrogate model for traffic optimization of congested networks: an analytic queueing network approach," *Techincal Report*, EPFL, 2009

[101] M. Baykal-Grsoy and Z. Duan, "Stochastic Models of Traffic Flow Interrupted by Incidents," *IFAC Symposium on Transportation Systems*, 2009

[102] H. Kobayashi, "Application of the Diffusion Approximation to Queueing Networks II: Nonequilibrium Distributions and Applications to Computer Modeling," *Journal of the Association for Computing Machinery*, vol. 21, no. 3, July 1974

[103] TomTom, "How TomTom's HD TrafficTM and IQ RoutesTM data provides the very best routing," *White paper*, 2010.

[104] Meihui TrafficCast, `http://www.meihuichina.com/` [Accessed in July 2012].

[105] R.W. Rosenthal, "A class of games possessing pure-strategy Nash equilibria," *International Journal on Game Theory*, vol. 2, no. 1, 1973.

[106] A. Byde, M. Polukarov, N. R. Jennings, "Games with congestion-averse utilities," *Algorithmic Game Theory*, vol. 5814, no. 1, 2009.

[107] J. Ferreira, P. Pereira, P. Filipe, and J. Afonso, "Recommender system for drivers of electric vehicles," *International Conference on Electronics Computer Technology (ICECT)*, 2011.

[108] H. Qin, W. Zhang, "Charging scheduling with minimal waiting in a network of electric vehicles and charging stations," *ACM VANET*, 2011.

[109] L. Zhao, S. Prousch, M. Hubner, A. Moser, "Simulation methods for assessing electric vehicle impact on distribution grids," *IEEE Transmission and Distribution Conference and Exposition*, 2010.

[110] L. Zhao, P. Awater, A. Schafer, C. Breuer, A. Moser, "Scenario based evaluation on the impacts of electric vehicle on the municipal energy supply systems," *IEEE Power and Energy Society General Meeting*, 2011.

[111] O. Sundstrom, C. Binding, "Flexible charging optimization for electric vehicles considering distribution grid constraints," *IEEE Trans. on Smart Grid*, vol. 3, no. 1, Mar. 2012.

[112] Q. Gong, Y. Li, Z.-R. Peng, "Trip-based optimal power management of plug-in hybrid electric vehicles," *IEEE Trans. on Vehicular Technology*, vol. 57, no. 6, Nov. 2008.

[113] F. Malandrino, C. Casetti. C.-F. Chiasserini, "A game-theoretic approach to EV driver assistance through ITS," *IEEE PIMRC*, 2012.

[114] K. Barry, "Battery-boosted quick-charger doesn't shock system," *WIRED*, Mar. 2012, `http://www.wired.com/autopia/2012/03/battery-boosted-rapid-ev-charging-station-wont-shock-\the-electric-grid/` [Accessed in July 2012].

[115] A. Fabrikant, C. H. Papadimitrou, K. Talwar, "The complexity of pure Nash equilibria," *ACM STOC*, 2004.

[116] OpenStreetMap, `http://www.openstreetmap.org/` [Accessed on July 2012].

[117] N. Nisam, T. Roughgarden, E. Tardos, V. V. Vazirani (Eds.), *Algorithmic game theory*, Cambridge University Press, 2007.