Avoiding the Gridlock

—

Information Dissemination in Vehicular Networks

Inaugural-Dissertation

zur

Erlangung des Doktorgrades der

Mathematisch-Naturwissenschaftlichen Fakultät

der Heinrich-Heine-Universität Düsseldorf

vorgelegt von

Christian Lochert

aus Mannheim

Düsseldorf, Oktober 2008

Aus dem Institut für Informatik
der Heinrich-Heine-Universität Düsseldorf

Gedruckt mit der Genehmigung der
Mathematisch-Naturwissenschaftlichen Fakultät der
Heinrich-Heine-Universität Düsseldorf

Referent: Prof. Dr. Martin Mauve
Heinrich-Heine-Universität Düsseldorf

Koreferent: Prof. Dr. Hannes Hartenstein
Universität Karlsruhe (TH)

Tag der mündlichen Prüfung: 24. November 2008

Zusammenfassung

In dieser Arbeit wird untersucht, wie sich Informationen in einem aus Fahrzeugen bestehenden, drahtlosen Ad-Hoc Netzwerk verbreiten können. Fahrzeuge kommunizieren mittels entsprechender Funktechnologie wie beispielsweise IEEE 802.11 direkt miteinander und bilden dadurch ein so genanntes *Vehicular Ad-Hoc Network (VANET)*. Fahrzeuge machen autonom Beobachtungen über die aktuelle Straßenlage. Diese Beobachtungen werden von vielen Fahrzeugen zu vielen Fahrzeugen geschickt. Es entsteht somit eine n-zu-m-Kommunikation mit n Sendern und m Empfängern, wobei sowohl die Beobachtungen als auch die Sender und Empfänger über die Zeit veränderlich sind. Das Ziel dieser Arbeit ist es, Methoden zu entwickeln, um diese Informationen an die Teilnehmer des Netzwerks zu verteilen. Bei diesen können sie von einem Navigationssystem als Eingabe für die Routenberechnung verwendet werden. Wichtige Fragestellungen betreffen zum einen die Reichweite des Informationsaustauschs sowie zum anderen die Geschwindigkeit, mit der sich Informationen ausbreiten können.

Ein wichtiges Hilfsmittel bei der Analyse von Mechanismen und Anwendungen für VANETs stellt die Verwendung von Simulatoren dar. Aufgrund der immensen Anforderungen an diese Simulatoren, die Realität so genau wie möglich abzubilden, dabei aber eine möglichst hohe Effizienz zu erreichen, existieren nur einige wenige Spezialsimulatoren. Diese dienen beispielsweise dazu, die Simulation von Daten und Signalen in Netzwerken oder die Bewegung von Fahrzeugen in Städten oder auf Autobahnen zu modellieren. Im ersten Schritt dieser Arbeit wird gezeigt, wie durch Kopplung einzelner Simulatoren ein Meta-Simulator erstellt werden kann. Dieser Simulator benutzt die (Teil-)Ergebnisse eines Spezialsimulators als Eingabe für den jeweils anderen Simulator. Dadurch ist es beispielsweise möglich, die Geschwindigkeit einzelner Fahrzeugen (in einem Vehrkehrssimulator) in Abhängigkeit von erhaltenen Datenpaketen (eines Netzwerksimulators) zu beeinflussen. Die Simulatorkopplung besteht aus dem Netzwerksimulator ns-2 sowie dem Verkehrssimulator VISSIM und stellt das zentrale Evaluationswerkzeug für alle in dieser Arbeit vorgestellten Protokolle und Algorithmen dar.

Darauf folgend werden die beiden zentralen Herausforderungen der Informationsverbreitung in solchen Netzwerken formuliert: *i) beschränkte Konnektivität* und *ii) beschränkte Bandbreite*. Die erste Herausforderung betrifft die Fragestellung, wie sich Informationen generell und mit welcher Geschwindigkeit in einem VANET ausbreiten können. Von elementarer Bedeutung hierbei ist, dass die verwendete Technologie noch

nicht weit verbreitet ist. Insbesondere während eines Einführungsszenarios wird die Penetrationsrate der Fahrzeuge, die mit Geräten zur Kommunikation ausgestattet sein werden, sehr gering sein. Bezogen auf das untersuchte Ad-Hoc Netzwerk lässt dies darauf schließen, dass das Netz durch Partitionierung in viele einzelne Teile aufgespalten sein wird. Durch diese Beschränkung ist eine schnelle und zuverlässige Verbreitung von Informationen nicht gewährleistet. Basierend auf dieser Erkenntnis wird das Konzept der Stützstellen vorgestellt. Diese stellen zusätzliche Infrastruktur für das ansonsten rein kooperative, selbständige Netzwerk dar. Sie bilden gewissermaßen ein Rückgrat für das allein auf Fahrzeugen basierende VANET. Da der Einsatz dieser Geräte mit zusätzlichem Aufwand verbunden ist, wird untersucht, welche Eigenschaften diese Stützstellen besitzen müssen. Weiterhin wird analysiert, wie viele Stützstellen mindestens nötig sind, um die Informationsverteilung positiv zu beeinflussen. Anhand einer Beispielapplikation wird gezeigt, wie bereits mit Hilfe weniger Stützstellen eine gute und schnelle Informationsverteilung bei sehr geringer Penetrationsrate über große Strecken hinweg gelingen kann.

Durch die vielen existierenden Datenquellen, die mit zunehmender Netzwerkgröße weiter anwachsen, nimmt auch der Umfang der zu verteilenden Daten stark zu. Um die zweite Herausforderung der beschränkten Bandbreite anzugehen, wird untersucht, auf welche Weise die Datenmenge zunimmt. Es werden Schranken für Verfahren zur Informationsverteilung bestimmt, um die Kapazität des Netzwerkes nicht zu stark zu beanspruchen. Diese Schranken begründen den Einsatz von Aggregationsverfahren, die Informationen mit zunehmender Distanz zusammenfassen, und zeigen gleichzeitig wie stark die Aggregation der Daten über eine bestimmte Distanz durchgeführt werden muss.

Aufbauend auf diesen Erkenntnissen wird ein Verfahren vorgestellt, welches diese beiden Herausforderungen bewältigt. Insbesondere wird ein Aggregationsverfahren für Stadtszenarien präsentiert. Diese Szenarien sind gekennzeichnet durch ihre Zweidimensionalität der zu betrachtenden Segmente bzw. Straßen. Informationen über Straßenabschnitte werden sinnvoll zusammengefasst und über weite Strecken verteilt. Die zentrale Eigenschaft dieses Aggregationsmechanismus besteht aus einer mehrschichtigen, hierarchischen Aggregation basierend auf dem *Landmark*-Prinzip. Um dem Problem der geringen Ausstattungsraten zu begegnen, wird ein Optimierungsverfahren vorgestellt, um möglichst gute Platzierungen für die oben genannten Stützstellen zu finden. Da der mögliche Lösungsraum sehr schnell sehr groß werden kann, wird ein genetischer Algorithmus für die Lösung des Optimierungsproblems benutzt. Durch die prototypische Implementierung eines Navigationssystems wird die Vorteilhaftigkeit des Aggregationsverfahrens in Zusammenspiel mit der optimierten Stützstellenplatzierung gezeigt.

Aggregationsverfahren bewältigen allerdings nicht nur die genannten Herausforderungen, sondern sind auch Ursache für eine weitere Herausforderung, die in Verbindung mit dem Verschmelzen mehrerer Aggregate auftritt. Wenn ein Knoten zwei Datenpakete empfängt, die denselben Bereich beschreiben, muss entschieden werden, welches Aggregat „bessere“ Informationen beinhaltet. Standardmäßig werden daher Zeitstempel vergeben, um die Aktualität der Aggregate und ihrer Inhalte zu kennzeichnen. Falls die Aggregation allerdings auf Flächen beruht und ein Knoten mehrere Aggregate mit sich teilweise überlappenden Gebieten erhält, kann mittels deterministischer Standardansätze nicht entschieden werden, wie diese Informationen zusammengefasst werden können. Im Ergebnis wird entweder ein falsches Aggregat berechnet oder es kann nur eines der beiden Aggregate für die weitere Verwendung herangezogen werden. In dem letzten Beitrag dieser Arbeit wird als Alternative dazu ein probabilistisches Verfahren vorgestellt. Die zentrale Eigenschaft dieses hierarchischen Aggregationsverfahrens ist, dass Aggregate implizit zusammengefasst werden können. Das neue Aggregat wird immer die Vereinigung der beiden alten Aggregate darstellen. Dabei ist es weder notwendig, dass sich Gebiete überlappen noch müssen die einzelnen Inhalte der Aggregate bekannt sein.

Abstract

In this thesis we analyze information dissemination in the context of vehicle-to-vehicle communication. Cars can communicate with each other by using radio technologies like IEEE 802.11. They implicitly form a so called Vehicular Ad-Hoc Network (VANET). Vehicles autonomously make observations about the current traffic situation of a road. In order to allow nodes to create an overview of the whole scenario these observations should be sent by many vehicles to many other vehicles probably far away. One goal of this thesis is thus to analyze the process of information dissemination as well as to provide mechanisms for the information dissemination. A navigation system may then use this data to calculate the fastest route based on the current traffic situation. In order to meet the requirements of such a system the information has to be transmitted in a timely fashion. It is furthermore important to reach distant regions to inform other vehicles in due time.

In the analysis of mechanisms and applications for VANETs simulators are an important building block. However, it is a hard challenge to model the reality as precise as possible while gaining high efficiency during the execution of simulations. Only special simulators developed for one single purpose are able to deal with these demands. These are, e. g., tools for the simulation of data networks or for the movement of vehicles on city roads or highways. In the first contribution we present results of coupling different specialized simulators. We develop a meta-simulator environment which allows one simulator to interact with another simulator by exchanging (partial) results or to react upon the input of these results. We are thus now able to analyze the implications of information dissemination in a network for instance on the movement of vehicles in a given scenario. In the following chapters we will use this coupling—consisting of ns-2 for the simulation of data packet networks and VISSIM for the simulation of car movement—for the evaluations of the mechanisms and algorithms proposed in this thesis.

The following contribution poses in detail the two major challenges of information dissemination in VANETs: *i) limited connectivity* and *ii) limited bandwidth.* The first challenge corresponds to the general feasibility of information dissemination in a VANET and its dissemination speed. Especially during the rollout of this technology this is an important aspect to consider. In an early stage the penetration ratio of vehicles equipped with VANET devices will be low. Regarding the ad-hoc network, it becomes obvious that a lot of partitions will exist that hamper a proper, fast and reliable spread of data. In order to deal with these limitations during the rollout phase we introduce the concept of (stationary) supporting units. These supporting units represent additional infrastructure devices. They build a backbone for the VANET which is formed by vehicles and their ad-hoc communication. Due to the additional costs we analyze the minimum necessary deployment of these units. We further demonstrate by the means of an example application that only few of these supporting units are needed to improve the performance of information dissemination significantly by spanning distant regions in a timely fashion.

By using multiple vehicles as data sources as well as with an increasing size of the network the amount of data, that is to be disseminated, grows significantly. In order to tackle the second challenge—limited bandwidth—we analyze how this increase in the amount of data looks like. We determine lower and upper bounds for the considered limitations of the bandwidth. These bounds motivate the usage of aggregation mechanisms. The main task of these mechanisms is to subsume information in relation to an increasing distance.

Following these insights we present an approach which tackles both mentioned challenges. In particular, we present an aggregation mechanism dealing with the additional challenge of a city scenario. This specific type of scenario is defined by its two-dimensional characteristics of the considered topology or streets, respectively. By using this aggregation mechanism information about street segments can be subsumed in an efficient and meaningful manner. The algorithm is implemented by a multilayered and hierarchical aggregation based on the *landmark* principle. In addition, we present an optimization method for the proper placement of supporting units in order to deal with the low amount of equipped vehicles. Due to the rapid increase of the possible space of solutions we propose to make use of a genetic algorithm in order to solve this hard optimization problem. By implementing a prototype navigation system we underline the performance of the aggregation scheme in combination with the placement of supporting units.

A distinct challenge of aggregation mechanisms is the merging of aggregates. For instance when one node receives two different aggregates describing the same area it has to decide which one of them contains “better” information. A standard approach is to use timestamps to indicate the up-to-dateness. By applying sophisticated mechanisms this problem might be solved. However, if the aggregation is based on areas and one vehicle receives two aggregates describing overlapping areas there are no (deterministic) standard approaches how to subsume these aggregates. The result is either a wrong aggregate or only one aggregate might be used further on. The last contribution of this thesis constitutes a probabilistic approach for the representation of aggregates. The central property of this hierarchical scheme is that overlapping aggregates can be subsumed implicitly. It is therefore not necessary to know the distinct entries of the aggregate. We show by means of an example application the benefit of this approach.

Acknowledgements

This thesis is the result of my work as a research scientist at the Computer Networks Research Group at the Heinrich Heine University Düsseldorf. However, it would not have been possible without the support and assistance from a lot of people I would like to express my deepest gratitude.

First of all I am grateful to my advisor Martin Mauve for asking me to join the Computer Networks Research Group and giving me the opportunity to do my PhD. During the last years he accompanied me as my supervisor and supported me throughout the whole research and finishing period. I also like to thank my referee Hannes Hartenstein. He not only agreed to review my thesis but he also was my supervisor during my diploma studies. It was due to him I experienced what research is and who prepared the ground for my PhD studies later on.

I like to thank my colleagues at the chair, Björn Scheuermann, Wolfgang Kieß, Jędrzej Rybicki, Yves Igor Jerschow, Michael Stini, and Tran Thi Minh Chau for the technical discussions and their most valuable feedback on my rough ideas. Especially, I appreciate the collaboration with and cooperativeness of Björn Scheuermann during the research on two major projects. I also owe my gratitude to my "external" co-authors of my papers. I'd like to highlight the lively discussions with Holger Füßler and Matthias Transier.

The student helpers in the projects I have been working on also deserve my deep gratitude. In particular, I want to mention Andreas Barthels, Alfonso Cervantes, Markus Koegel, Emilio Janzen, and Michael Singhof. I also thank Marga Potthoff, our group's secretary, for "fighting against the bureaucracy monster", and Christian Knieling for providing all the software and configurations I needed to perform my simulation studies.

During the period of doing research it is not less important to have a rest from time to time. Therefore, my thanks go to the people from the "IGFZS": Cristian Pérez de Laborda, Andrea Führer, Christopher Popfinger, Evguenia Altareva, Alexei Rubinstein, Michael Stini, and Wolfgang Kieß as well as to Jens Schwarzwälder and Anne Kerres.

I dedicate this thesis to my parents, Brigitte and Manfred Lochert. They always supported me and encouraged me to be an open-minded person. I would not have achieved anything without them. Finally, this thesis is especially dedicated to Dagmar Striedinger. She constantly supported me with her love and affection. Even during burdensome times she found the right words to set me back on the right track.

Contents

List of Figures

List of Tables

List of Abbreviations

ACM	Association for Computing Machinery
BNF	Backus-Naur Form
CDF	Cumulative Distribution Function
COM	Component Object Model
CORSIM	CORridor SIMulation
DLL	Dynamic Link Library
DSRC	Dedicated Short-Range Communication
DTN	Delay Tolerant Network
FM	Flajolet Martin
GA	Genetic Algorithm
GPRS	General Packet Radio Service
GPSR	Greedy Perimeter Stateless Routing
GSM	Global System for Mobile Communications
ID	Identifier
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
ITS	Intelligent Transportation Systems
LoS	Line of Sight
MATLAB	MATrix LABoratory
MPGA	Multi-Population Genetic Algorithm
ns	Network Simulator
PCSA	Probabilistic Counting with Stochastic Averaging

RSU	Roadside Unit
SSU	Stationary Supporting Unit
SU	Supporting Unit
SUMO	Simulation of Urban MObility
Tcl	Tool Command Language
TCP	Transmission Control Protocol
TTL	Time To Live
UMTS	Universal Mobile Telecommunications System
VANET	Vehicular Ad-Hoc Network
VISSIM	Verkehr In Städten – SIMulation
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network

Chapter 1

Introduction

In our daily life we experience that the amount of cars on the roads is increasing more and more. The consequences are traffic jams and accidents. These cause hundreds of thousands of injured people. According to the German Federal Statistical Office, in 2007 there happened 2.3 million accidents with more than 400 000 injured people and almost 5 000 fatalities solely in Germany [Deu08, Bun08]. Summed up the numbers for the European Union, almost 45 000 people lost their life in 2004 [BM07]. On the other hand in times of increasing prices for oil and gas these jams cost a lot of money and incidentally cause the pollution of our environment. One reason amongst others is that drivers do not possess information that they would have needed to react properly and in good time to avoid an accident or to bypass a traffic jam.

At the same time more and more navigation systems are sold that aim to help the driver to find the fastest route to a given destination. These systems are, however, neither able to identify traffic jams nor any other incidents affecting the conditions of a road. They are thus not able to react on dynamic road conditions in order to find a current optimal route.

This thesis is on sharing topical spatio-temporal information between cars. Observations on the present road or traffic situation are made autonomously by vehicles and shared with other vehicles in an ad hoc fashion. The information is spatial because it is an observation about a distinct location or area and it is temporal because the value can vary during the time. One example of a spatio-temporal information is the status of an arterial road of a city which is heavily used during rush hour. In the off-hour the road is used by comparatively few cars only. We analyze how this information can be spread over relatively large distances to inform drivers of cars in due time. In the following chapters we assume that cars are equipped with dedicated short-range communication (DSRC) [DSR] devices like WLAN respectively IEEE 802.11 [IEE07] and are thus able to transmit and receive data packets.

A lot of effort has been put into a common standardization approach across a range of governments and car manufacturers. For instance in the US, a distinct frequency band for car-to-car communication has been reserved. These frequencies should then be used by specialized radio equipment according to IEEE 802.11p [IEE08]. Based on this technology car manufacturers can install vehicle-to-vehicle communication devices in order to share information among many vehicles that can be used by a traffic information system.

Capacity restrictions implying a limited bandwidth as well as low equipment ratios of vehicles, however, prevent simple solutions. Considered that each vehicle is the source of information which is transmitted over possibly large distances, it becomes obvious that the limited capacity of the network will not be sufficient. A further challenge for this information sharing paradigm is that during the rollout phase of this technology only few cars will be equipped with VANET technology. This inhibits a rapid dissemination of these observations and affects the results of the information system. In the following chapters we will further elaborate on these two challenges and present approaches to tackle it.

The remainder of this thesis is structured as follows.

In Chapter 2, we investigate related work. We look especially at different approaches to inform the driver of a recent incident or the current traffic situation of a road. Most of these approaches use a simplistic form to spread information to other regions or drivers. We describe different existing methods for the dissemination and classify the ideas and paradigms of these approaches. Aggregation techniques for summarizing information have also been proposed. We introduce these methods and classify their basic ideas similar to the dissemination approaches.

Following this chapter, in Chapter 3 tools for the simulation of inter-vehicle scenarios are presented. In general, pre-calculated traces of vehicular movement patterns or predefined communication traces are used for the design of a simulation study. But this static approach is insufficient for the evaluation of VANET applications like the transmission of emergency warnings. To deal with a received warning message, vehicles should be able to adapt their speed so as to prevent an accident. With the static input direct reactions are not possible. We show how multiple specialized simulators can be interlinked at runtime to perform realistic simulations for vehicular (ad-hoc) networks. The interlinking combines a network simulator (ns-2), a traffic simulator (VISSIM), and an application level simulator (Matlab/Simulink) to enable the realistic analysis of VANET applications. We present the architecture, show how we have implemented it, and discuss performance issues. By the means of a simple example application we demonstrate the functionality of the simulator coupling. In order to simulate the radio propagation as realistic as possible while sustaining the abstraction level we propose an obstacle model that emulates the blocking of radio signals through the walls of buildings. Last, we show how to influence the movement of cars controlled by the interlinked network simulator.

In Chapter 4, we analyze the feasibility of information dissemination approaches using inter-vehicle communication. Here, we consider information dissemination in vehicular ad-hoc networks (VANETs) in city scenarios. A particular focus of this study is on the rollout phase when only few cars participate in the system. After analytical considerations, we focus on simulations using a detailed model of a whole city. We assess the dissemination performance depending on the amount of equipped vehicles on the road. For few equipped vehicles, we show that dissemination speed and coverage will not be sufficient. Therefore, we propose to use specialized, but simple and inexpensive infrastructure: Stationary Supporting Units (SSUs). If a small number of SSUs is installed in a city and connected via a backbone network, the dissemination performance improves dramatically, especially during the VANET rollout phase. Thus, SSUs foster a faster and earlier rollout of working, dissemination-based VANET applications.

Capacity restrictions of the network are considered in Chapter 5. We analyze how the increasing amount of data which needs to be transmitted contrasts with the available bandwidth. In consequence to these considerations we show that aggregation techniques are needed in order to disseminate the data to remote parts of the considered scenario. We pose requirements for these techniques and show formally the validity of these demands.

In Chapter 6, we present an information dissemination protocol based on deterministic aggregation. We investigate how a VANET based traffic information system can overcome the two key problems of strictly limited bandwidth and minimal initial deployment. First, we present a domain specific aggregation scheme in order to minimize the required overall bandwidth. Then, we propose a genetic algorithm which is able to identify good positions for static roadside units (Stationary Supporting Units) in order to cope with the highly partitioned nature of a VANET in an early deployment stage. A tailored toolchain allows to optimize the placement with respect to an application-centric objective function, based on travel time savings. By means of simulation we assess the performance of the resulting traffic information system and the optimization strategy.

In contrast to the previous chapter where the focus was on the hierarchical aggregation of data, in Chapter 7, we concentrate on the *representation* of aggregates. In order to merge overlapping aggregates efficiently we propose a probabilistic aggregation scheme. We present an algorithm for the hierarchical aggregation of observations in dissemination based, distributed traffic information systems. Instead of carrying specific values—e. g., the number of free parking places in a given area—our aggregates contain a modified Flajolet-Martin sketch (FM sketch) as a probabilistic approximation. The main advantage of this approach is that the aggregates are duplicate insensitive. This overcomes two central problems of existing aggregation schemes for VANET applications. First, when multiple aggregates of observations for the same area are available, it is possible to combine them into an aggregate containing all information from the original aggregates. This is fundamentally different from existing approaches where typically one of the aggregates is selected for further use while the rest is discarded. Second, any observation or aggregate can be included into higher level aggregates, even if it has already been added—directly or indirectly—before. As a result of those characteristics the quality of the aggregates is high, while their construction is very flexible. We demonstrate these traits of our approach by a simulation study.

Finally, we conclude this thesis in Chapter 8 with a short summary. By highlighting the major building blocks we reflect the achieved contributions.

Chapter 2

Car-to-Car Communication

In the last ten years, increasing research effort has been put into the area of Intelligent Transportation Systems (ITS) [ITS] as well as Vehicular Ad-Hoc Networks (VANETs). Vehicles are equipped with radio communication devices and if they are within communication range, they are able to communicate directly with each other in an ad hoc fashion. In contrast to Mobile Ad-Hoc Networks (MANETs) questions of power supply and energy efficiency are less relevant in VANETs. However, due to the potential high (relative) speeds of vehicles new challenges need to be solved. In 2000 one of the first projects that dealt with car-to-car communication in Germany was the FleetNet project [Enk03]. The project aimed to assist the driver by, e. g., transmitting accident warnings or requesting the current traffic status ahead. In this context most of the research effort concerned network-centric aspects like data routing.

After the development of technologies and procedures for the direct communication between vehicles a number of applications has been proposed, too. These make use of vehicular ad-hoc communication to increase driving comfort or safety. Especially applications that aim to increase the safety of a driver play an important role. In order to foster these research efforts many car manufacturers as well as governments have come together and tackle these tasks in several projects [NoW, PAT, PRe, DSR, C2C]. The perspective of research has thereby been broadened to additional applications. Comfort applications for VANETs have been discussed. These deal with the flow of traffic for instance, and often they use some form of data dissemination. Thus, in this context, our contributions here are of immediate relevance. In this chapter, we describe the different approaches dealing with information dissemination in VANETs. We show how they approach with the problems considered in this thesis and relate them to our work. Table 2.1 depicts a summary of the attributes of a vehicular network.

Table 2.1: Characteristics of a VANET and its applications.

Type	Characteristics
Specific goal of the application	• Safety • Comfort
Type of transmission	• Unicast • Multicast • Broadcast • Geocast • Flooding
Participants of the network	• Vehicles only • Vehicles plus infrastructure • Infrastructure only
Requirements of the network	• Time constraints • Reception reliability
Type of propagation	• Direct communication vehicle-to-vehicle (V2V) • Communication via infrastructure vehicle-to-infrastructure (V2I) • Hybrid approach vehicle-to-X (V2X) • Indirect by the locomotion of vehicles
Requirements of the nodes	• Network or routing centric • Application centric

2.1 Security

When speaking about applications for VANETs in a real environment questions about security arise. Especially for safety applications it is vital that only accidents that really happened are being reported in a warning message. But even in comfort applications it is necessary to assure that only correct information on, e. g., the average speed on a road is measured and transmitted.

A lot of studies thus analyze how security mechanisms can be used to avoid faked messages that corrupt the procedure of an application. For instance, it has been studied how VANETs and aggregated messages in them can be made tamper-proof or resistant to cheaters [GGS04, JW04, RAH06, PRGI06, PZ08, IW08b]. Since all these questions are largely orthogonal to the fundamental challenges how to disseminate messages and how aggregates should be built, compared, and merged, their ideas can be combined with the approaches of this thesis.

2.2 Routing vs. Information Forwarding

For the transport of information in a network routing protocols provide paths to forward data packets towards a destination. It is thus consistent to analyze existing (unicast) routing protocols in the context of VANETs. It has been worked out that these routing techniques—which have been developed for Mobile Ad-Hoc Networks (MANETs)—were not able to react sufficiently fast to the high mobility of nodes in VANETs. The main reason was that a lot of control messages need to be exchanged between nodes in order to map the topology. Due to the rapid change of the topology control messages are sent again which is repeated constantly.

It was one major contribution to use position based routing approaches. The basis of all these protocols is that they do not establish a topology based unicast route between one sender and one receiver [MWH01]. Instead, these approaches make use of the geographic position of nodes. Using simple greedy approaches protocols like Greedy Perimeter Stateless Routing (GPSR) [KK00] or face-2 [Sto02] forward packets to a destination based on current local information at one node along the "route". The number of hops and the identity of forwarders might change from packet to packet. Nevertheless, it remains a problem to find out at which geographic position the destination can be located. Surveys which summarize the routing approaches of the last years can be found, e. g., in [CVCAC06] or in [LW07].

As we have seen in Table 2.1 the unicast communication paradigm, however, is only one out of many paradigms to transmit data. Unicast communication implies that there are only two nodes involved. Considering VANET applications which require that many nodes do receive a message, the unicast approach does not seem to be suitable. Rather, data should be disseminated to the relevant receivers. If we consider a WLAN-like network, transmissions are, due to the shared medium, always performed in a broadcast fashion. Instead of limiting the number of receivers artificially, protocols that forward messages should thus exploit this property.

2.3 Information Dissemination

While VANET safety applications typically require information only from a relatively limited geographical area—the vicinity of the car—, this is not the case for many convenience and driving comfort applications. For example, navigation systems could make use of information on the current traffic situation in a larger surrounding. Another example are VANET-based parking guidance systems; these distribute information about the parking situation in an entire city. In this type of applications, information is cooperatively collected and shared within a large area, like a highway network or the road network of a city. The information might be on traffic intensity and travel times, free parking places, or any other parameter that can be observed by individual participants and collected and distributed by a cooperative mechanism in the vehicular ad-hoc network.

Distributing information over long ranges in a VANET is a very challenging task. It can be fulfilled only when the inherent properties and limitations of VANETs are considered in all components of the system. Particularly relevant are *i)* connectivity and *ii)* capacity constraints. Limited initial market penetration results in a very sparse network. Even once a higher penetration ratio is achieved, network connectivity will often be constrained, e. g., during low-traffic hours. Among other effects, limited connectivity directly affects the possible speed at which information can be transported over a VANET, and hence directly limits the up-to-dateness of the shared information that can be achieved.

Capacity constraints in VANETs are mainly due to the limitations of the shared wireless channel. It is intuitively evident that the capacity of the network will not suffice to provide detailed and continuously updated information on every small-scale geographical entity to all participants of the network. This is also underlined by known results on the capacity limits of wireless multihop networks, first and foremost the seminal results by Gupta and Kumar in [GK00]. The limited capacity necessitates mechanisms to reduce the amount of information that needs to be transported through the network, by summarizing and aggregating individual data items.

Four central challenges have been identified in the literature on long-range data dissemination in VANETs:

- How to *obtain* the information, i. e., how to make local observations that can form a data basis for the application?
- How to *transport* the information, that is, given that information is available at some point in the VANET, how is it delivered to interested parties in other parts of the network?

- How to *summarize* measurements made by distinct VANET participants, given that it is not possible to distribute all individual observations?
- How to *aggregate* information on larger geographical entities, considering the fact that network capacity constraints do not allow for an arbitrarily detailed picture of distant regions?

A basic sketch of a dissemination-based system is shown in Figure 2.1. Information is collected from internal and external sources (i. e., from sensors in the car and potentially also from roadside infrastructure) and is stored in a local knowledge base. The contents of this knowledge base is made available to the application, and is furthermore shared and exchanged with other cars by means of wireless communication. The amount of data exchanged between cars is reduced by applying summarization and aggregation techniques to the data in the knowledge base.

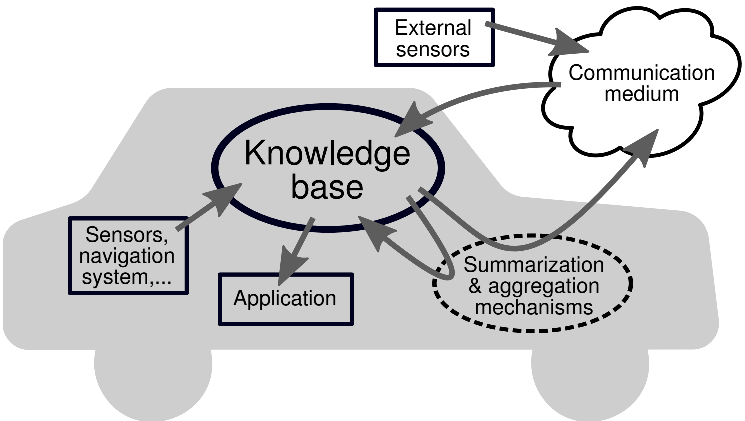


Figure 2.1: A dissemination-based VANET application.

Our discussion will roughly follow the structure given by the four questions posed above. The next section will look at issues of and proposals for local data generation, before we then consider protocol issues for data transport in Section 2.3.2. Sections 2.3.3 and 2.3.4 subsequently turn towards data representation and transformation aspects, for summarizing and aggregating the collected information, respectively.

2.3.1 Obtaining Local Measurements

Before a dissemination-based VANET application can start to propagate and process data, local observations (i. e., measurements of one or more local parameter(s)) need to be made. In the discussion of data dissemination protocols and dissemination-based applications, the aspect how these observations are made is often neglected—their existence is simply assumed, and the parameters to be disseminated are chosen in a more or less straightforward way. However, a closer look reveals that generating local observations that are suitable for specific applications is often not that straightforward, and that the choice of parameters to be transmitted in the network deserves careful consideration. It heavily depends on the specific purpose, i. e., on the aim of the application, and it is intimately interrelated with other design parameters concerning the protocol, aggregation mechanisms, etc.

Sensors in vehicles The easiest way to obtain information is to use data provided by integrated sensors in the vehicles. Many such sensors are already available and are used for other electronic subsystems in the car. They provide information on environment parameters like temperature, light intensity, position (GPS), speed, or road surface condition (ice, water, etc.). More and more often, sophisticated devices and processing logic are also able to provide information like the distance to preceding and successive vehicles (e. g., from radar distance measurements), visibility distance, or traffic signs and traffic lights (through camera pictures and image recognition techniques).

Reading the speedometer or rev counter alone can already provide important information and may allow to draw conclusions on the current traffic situation. In order to improve the accuracy and granularity of the information, it is possible to combine information from multiple sources, a process often called *sensor fusion*. Nowadays, many vehicle sensors are already connected to the Controller-Area Network (CAN), specified in [Int03, Int04], and sensor readings can be obtained in a standardized way via this channel. A system that integrates and combines data from different sources in order to gain insight into the current road situation is discussed by Ozbay et al. in [ONG07]. In the context of accurate localization, Boukerche et al. discuss sensor fusion in [BONL08].

The navigation system as an information source The majority of the VANET convenience applications discussed so far are more or less directly related to a navigation system. Prime examples are again a distributed traffic information system for finding routes with short travel times based on the current traffic situation and a system for finding free parking places. From the perspective of information generation in VANETs, the fact that more and more vehicles are equipped with a navigation system means that more and more vehicles have a particularly powerful and sophisticated kind of "sensor" at their disposal: a navigation system does not only have quite accurate position and speed information available, but also detailed map data and information about the intended driving direction. Unlike, for instance, a "bare" GPS receiver, a navigation system is therefore able to provide position information on a higher abstraction level: instead of geographical coordinates, a navigation system knows the road ID, driving direction, or even the lane (e. g., regular vs. turning lane).

Distributed traffic information systems like SOTIS [WER+03b] by Wischhof et al., TrafficView [NDLI04b] proposed by Nadeem et al., or CASCADE presented by Ibrahim and Weigle in [IW08a] often use vehicle position and driving speed as the underlying primitive (which is then exchanged in more or less aggregated form). If the aim of the application is to optimize the route choice based on the current traffic situation in the road network, an alternative approach is to instead use travel times along road segments as the local observations, as suggested by Goel et al. in [GIO04] as well by Xu and Barth in [XB06a]—with the aid of a navigation system, this parameter is straightforward to obtain locally. While individual (and quickly outdated) vehicle position information needs further, non-straightforward processing before it can serve as a basis for a road network routing algorithm, travel times along road segments can directly be used as an edge weight in road network representation of a VANET-supported navigation system.

Both approaches may exhibit their individual benefits and drawbacks. However, aggregation mechanisms, dissemination protocols, the interpretation of the values by the application, etc., differ largely depending on whether highly dynamic, short-lived individual vehicle positions and speeds or more time-stable, cooperatively estimated average travel times are shared in the network—the impact of the choice on other system components is evident in this example.

External sensors It is conceivable to complement the built-in sensors in the vehicles with additional, external information sources. These external sensors might, for example, reside in the road pavement and count the number of passing-by vehicles and thereby generate raw data for an analysis of the current traffic situation. Parking spaces can be equipped with devices to sense whether they are currently occupied or not, as presented in [PGP06] and [Mar08]. Often, it will not even be necessary to install new sensing equipment, because the data is already available. For example, parking meters may already know which parking places are occupied; induction loops are often present to sense passing-by cars; the electronics controlling traffic lights are aware of the current traffic light status.

Regardless whether new equipment is installed in order to supplement information collected by the cars themselves or whether readily available, external information sources are used, a way to communicate this data to the VANET is necessary. Typically, this will happen by integrating external devices into the wireless network, such that passing-by cars can communicate with them and thereby obtain the respective information. Concrete scenarios of such applications are, for example, discussed by Caliskan et al. in [CGM06, CBSM07].

The VANET device as a "sensor" An interesting idea to complement the so far mentioned car-internal and -external information sources is to draw conclusions about the traffic situation from observations concerning the VANET itself. Many proposed VANET safety applications build upon periodic beacons. Such information messages, single-hop broadcasted to all cars in the surrounding, can provide a fine-grained picture of the local environment, and therefore serve as a basis for an estimation of the local traffic density. Mechanisms in this direction are suggested, for instance, by Jerbi et al. in [JSRGD07]. Using VANET-based local information exchange to derive information about the local environment will, of course, be easier to implement once a substantial fraction of cars is equipped with VANET devices; until then, information acquired through periodic local presence announcements of VANET nodes will necessarily be incomplete.

Prediction of parameters All above mentioned sensed parameters take only the current situation into account. However, the authors of [CBSM07] point out that this will often lead to suboptimal results. They motivate this using a distributed application guiding cars to parking sites with free capacity. Due to a limited information propagation speed in the VANET, there will often be significant delay between the measurement and the arrival of the respective information at an interested car. Furthermore, a significant time span will typically elapse until the interested car is able to arrive at the parking place. A driver will, however, not be interested in the number of free parking places at some time in the past, but instead intends to maximize her individual probability that at least one parking place is free upon arrival at the parking site.

In their approach, Caliskan et al. assume that each parking lot consists of a number of parking places and has a central instance that is able to keep track about arrivals and departures (e. g., a fee payment terminal). They propose to not only monitor the current occupancy of each parking site, but also the current dynamics, expressed through an arrival rate and the average parking duration. With these additional parameters available, cars can make predictions about the probability to find a free parking place at the respective parking lot at the estimated time of arrival.

The central insight is of broader interest than just for a parking guidance application; similar observations of course also hold for other applications. The aspect discussed by Caliskan et al. therefore generally underlines that a holistic design of a VANET dissemination application makes sense: the nature of the information, the means by which it is obtained, the way it is used by the application, and the way it is communicated from its point of origin to interested parties are all interrelated.

2.3.2 Information Transport

After obtaining information from different sensors, a local view of a car's surrounding can be created. However, not only the car that obtained this information will need this data, but also other vehicles should be able to adapt their behavior based on this information. It is thus important to distribute the information to the cars interested in it. In the following paragraphs we deal with the different methods for the *transport* of information. In order to avoid misconceptions we define the terms *broadcasting, flooding,* and *beaconing* as follows. *Broadcasting* is a *single-hop* transmission of a packet to all nodes within radio range of the sending node. It is often used as a primitive to implement *flooding,* which means distributing a packet over a range spanning multiple wireless hops. The receiving nodes re-broadcast the packet and thereby deliver it to all nodes in the network or to a subset thereof, for instance the nodes within a limited geographical region. *Beaconing* is the periodic transmission of information to all neighbors within radio range. In a sense beaconing is periodic broadcasting. One single packet transmitted by beaconing is called a *beacon.*

Protocols for Information Transport

Flooding Well known from its frequent use in MANETs, one way to propagate information very fast is to use flooding. In a naïve implementation every node that receives this information will simply rebroadcast it. To avoid infinite packet duplication, each node will broadcast a given packet at most once. In addition a time to live (TTL) counter may be used to limit the area where the packet is distributed. This naïve approach will transmit a large amount of redundant packets, potentially leading to severe congestion. This is known as the "broadcast storm problem" originally identified in a study by Ni et al. in [NTCS99]. Many approaches have been proposed to deal with this problem.

In vehicular networks flooding is often used to disseminate traffic information messages to other vehicles. The general procedure is that a vehicle—the source—detects an incident or situation which should be communicated to other vehicles. The source thus starts flooding this information. In the following paragraphs we deal with different approaches that aim to limit the number of concurrent packets within the network. The common idea of all these methods is to influence the forwarding behavior of a forwarding vehicle, either by adapting the time when to forward the packet or by introducing rules on whether a given vehicle should forward the packet at all.

In the approach of Tonguz et al. it is proposed to adapt the broadcasts used to realize flooding depending on the density of the traffic [TWB+07]. When the traffic is dense not every car needs to retransmit the message. Alternatively, the broadcasting of information can also be adapted to the covered distance. In the closer vicinity of the source the message is repeated by many nodes. By overhearing ideally all or almost all nodes in close proximity to these broadcasters will be informed. With increasing distance to the source the frequency of broadcasts is decreased. Similar to this approach, in [BK06] Brønsted et al. propose to use flooding *only* in the surrounding of the message's source. This allows to inform vehicles close by very fast without having to flood a packet into a large area. Dornbush and Joshi rely in their study [DJ07] on techniques called "rumor spreading", "gossip", or "epidemic" dissemination. Here, a node which receives a message stores it locally and will forward it later to other nodes. Nevertheless, these techniques still use a flooding like approach. In order to limit the number of transmitted messages, nodes will decide to disseminate only the latest information. Also, approaches combining these two ideas are possible. In [WHF+07], Wegener et al. propose to adapt flooding not only to the density of nodes in the vicinity but also to the novelty of the flooded information. Thus, a new event will trigger the creation of a new message. To meet the density rate adaption criterion only few vehicles will further broadcast this newly generated information. While being forwarded the message gets older which results in a reduced broadcasting frequency. By limiting the number of concurrent packets this scheme aims to avoid the broadcast storm problem.

A slightly different scheme to limit the flooding of information is to let the receiver decide whether to rebroadcast the packet at all. Chisalita and Shahmehri, for instance, consider in [CS04] such an approach, albeit they primarily look at safety applications. They assume that when a dangerous situation is detected a warning message will be broadcasted. In contrast to simply rebroadcasting this message, a receiver of this data analyzes whether it is interested in the warning. This could be the case if the receiver is driving on the same road and in the same direction as the source vehicle of the message. The assumption then is that if it itself is interested in the data other cars in its vicinity might also profit from the message. The information is thus broadcasted again. This approach may be leveraged by including additional information in the message to ease the decision on which nodes are interested in the data. Ducourthial et al. present an approach in [DKS07] to add, e. g., geocast addresses to the message in order to facilitate the analysis performed by a receiving node. By evaluating this condition it may then decide whether to forward the message.

Similar to these schemes in [BSH00] Briesemeister et al. propose to take into account the distance to the previous forwarding node. The further away a node is, the more likely it will rebroadcast the received message. Nodes that overhear this transmission in between these two nodes do thus not need to broadcast it. So as to determine the furthest node explicitly Korkmaz et al. apply an RTS/CTS scheme for broadcasts in [KEÖÖ04]. After the transmission of a "request to broadcast" all nodes in transmission range send a jamming sequence called "black-burst" with a duration dependent on the distance between the sender and the node. The furthest node thus sends the longest burst and indicates by sending a "clear to broadcast" that it wants to receive this broadcast packet.

In order to reduce the overhead of such a scheme Zhao et al. use a position based geocast routing approach. In [ZZC07] they propose that the receiver decides—based on a local neighbor table—to which node the message will be forwarded. Due to the usage of "directed" broadcasts, nodes within the radio range can overhear the message. In an early study of Kosch et al. it is proposed to extend the reactive routing protocol AODV [PR99] likewise by some geocasting functionality as well [KSA02]. Messages containing information on, for instance, current road conditions are disseminated within a given region. The same authors also propose an approach where information is disseminated according to a local interest rate. If a node is interested in this data it will also forward the message. Although this approach is appealing because existing protocols can be used, it might have scaling limitations in large networks where a lot of nodes are the source of information. In order to deal with the distinct mobility patterns of vehicular networks Sormani et al. suggest in [STC+06] to add a "store and forward" mechanism to the receiving nodes. Thus, the nodes contribute to transport the information by wireless transmissions as well as by their own locomotion.

It should be noted that message spreading schemes where nodes decide whether to forward information based on their own interest miss the opportunity to use vehicles that are themselves not interested in the information for improved dissemination. This may turn out to be a severe performance issue. For instance, it has often been observed that using the oncoming traffic for information transport results in significantly better performance—the oncoming traffic will, however typically not be interested in data about the region it is coming from.

Williams and Camp aim in their study [WC02] to compare the different flooding approaches according to their performance and overhead. Although this study is done in the context of generic MANETs the results can be helpful for VANETs as well. Their results and conclusions underline the expectation that flooding approaches show very poor performance in terms of congestion. The study considered only one flooding "stream" starting at one single source. In a vehicular information dissemination scenario, information flooding might be started by multiple nodes concurrently which would most likely deteriorate the presented results further.

Request/Reply The transport of information in the context of vehicular applications is not limited to solely proactive flooding approaches. If the information is not needed by many users it is worthwhile thinking about reactive or on-demand algorithms. In this case a vehicle will ask explicitly for specific information by transmitting a request message. The final destination of this packet may be known, for instance if the user requests the current gas prices from the next gas station or asks a parking meter for a free parking spot. The forwarding of this message may be implemented in different ways.

Zhao and Cao propose in [ZC06, ZC08] to forward request messages in a unicast fashion by constructing an explicit route, while Basu and Little propose in [BL02, BL04] use position based routing approaches. The reply message will then be returned similarly to the request. Instead of asking for the information directly at the source, vehicles may also ask other vehicles passing by whether they know about it. According to Sago et al. the waiting time at a red traffic light may last long enough to fill up the local knowledge base by querying other cars [SSHN07]. In most studies requesting is performed in an indirect fashion by announcing the entries of the local knowledge base. A node that receives this message can analyze if it is aware of information that is new or unknown to the requester. Different techniques are proposed to implement these request mechanisms. In Figure 2.2 an autonomous update process is depicted. In addition to this procedure it is also possible to announce the local knowledge while other vehicles request explicitly for an specific entry. In contrast to a unicast request-reply mechanism, Wegener et al. propose in [WHF+07] a periodic single-hop broadcast requesting technique based on a data structure similar to the one of the study mentioned above. Furthermore, by using techniques to prevent overload of the network they suggest to adapt the frequency of these requests depending on the traffic density.

In order to overcome the partitioning in VANETs Fujiki et al. propose in [FKUH07] to make use of delay tolerant routing approaches. In this context request packets do not need to follow a prebuilt unicast route but are geocasted to a certain region and can be transmitted implicitly by the locomotion of vehicles. By doing this, different partitions can be spanned. The transport of the reply message follows the same rules as the request packets did.

It is worth stating that request/reply schemes face serious scalability problems. Due to the well known capacity constraints in wireless multihop networks, as stated by Gupta and Kumar in [GK00], the capacity usable by any source converges to zero as the size of the network increases to infinity. Thus these approaches must take special care to limit the request/reply exchange to vehicles that are close to each other or to employ some other way to limit the load on the network.

Clustering In some situations it becomes obvious that flooding a message into a whole network is not appropriate and will cause a high level of congestion. At the same time request/reply schemes might not be appropriate, e. g., since the information is needed by many nodes at the same time. It has therefore been proposed to replace the unstructured flooding of packets by some sort of hierarchical distribution using clustering.

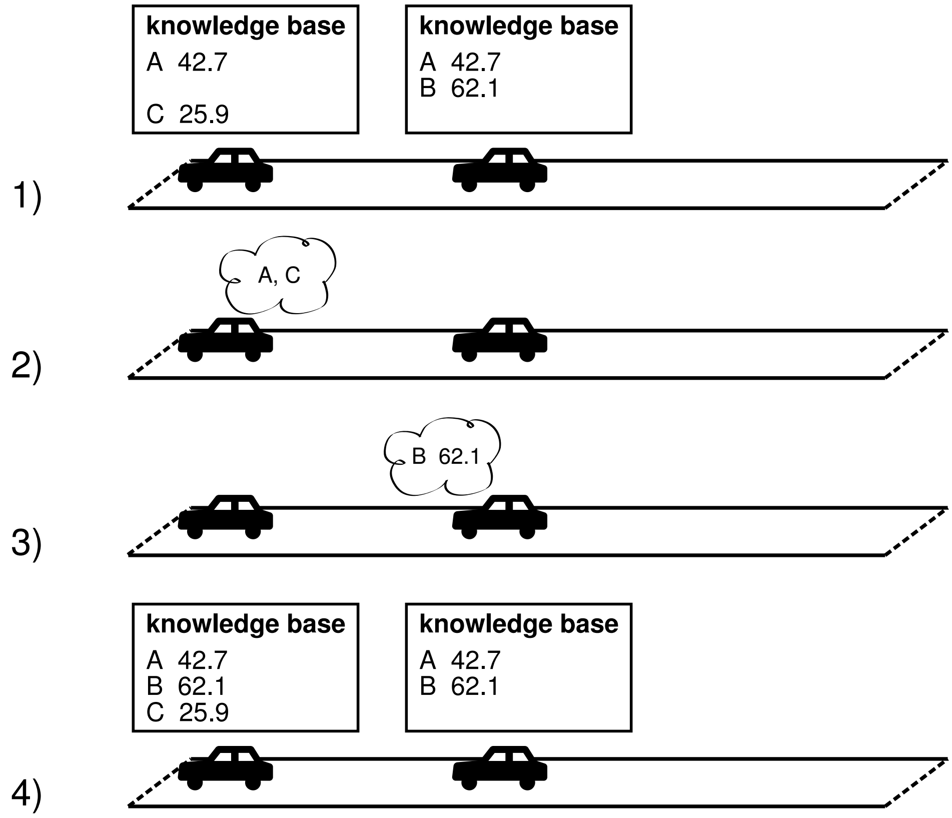


Figure 2.2: Implicit information exchange. In 1) two cars with different knowledge bases meet. In 2) the left car propagates its own knowledge. By comparing this announcement the right car can differentiate which entries it owns and are thus needed by the left car. In 3) it transmits the missing entry. Finally, in 4) the left car has updated its knowledge base.

Little and Agarwal present the Directional Propagation Protocol (DPP) [LA05, ASL07, ASL08]. In this approach the authors consider information dissemination along a highway. When the traffic is dense nodes within some region form a cluster. Within this cluster messages are shared with all nodes of this cluster. In order to propagate a message in one direction overlapping clusters are used to transmit the message. Clusters in the opposing direction are considered as well to propagate the message. DPP adopts procedures of a Delay Tolerant Network (DTN). If there is no overlapping cluster the propagation of messages is done simply by the locomotion of vehicles.

A dissemination approach that also makes use of clustering techniques is proposed by Chang et al. in [CLC08]. If a car needs some information it forms a cluster. This initiates the clustering process. In order to transmit data to other clusters relay vehicles connect two clusters. In a first phase information within one cluster is gathered using a Time Division Multiple Access (TDMA) fashion. After the first—collection—phase all cluster heads own the information of all vehicles within their cluster. In the second—retrieval—phase this information is then sent back to the initiator of the first cluster, i. e., the requesting node.

Another cluster based approach is presented by Chennikara-Varghese et al. in [CVCHO07]. Here, cars within a small region autonomously form a cluster. Due to the close vicinity direct communication between those cars is possible. However, communication to other vehicles or clusters has to be performed by relaying vehicles that are part of several clusters. In the context of parking meter status dissemination Basu et al. propose in [BL02, BL04] to use clustering approaches only within the group of the static parking meters.

Sharing In [LM07] Leontiadis and Mascolo present an approach which is based on the publish/subscribe paradigm. Messages are disseminated and kept in a predefined area. If a node is interested in any type of message it can subscribe to this event locally. If a messages is received by that node it can easily check if this matches its subscription or not. Information is duplicated and these replica are transmitted to some nodes within this area. If a node leaves this area the information is then transferred to a node which moves into the area or remains within this area. The authors show in their study that the number of duplicates can be dynamically adjusted and is in general lower than the number of subscribers. The main difference of this approach compared to the approaches above is that information should remain in some area.

The approach proposed by Shinkawa et al. in [STK+06] follows the same general idea. Vehicles possess information about their current area. If they leave this area the information is discarded. But in order not to loose this information globally, especially if the density of equipped nodes is low, the authors propose to use buses as "message ferries" which will not drop any received information. The motivation behind using these vehicles is that they follow a regular and recurrent route covering multiple regions and are thus able to supply other vehicles with information.

Beaconing In many VANET applications, time-varying data is announced periodically by each node in form of a beacon. In this way the network load is limited to a fixed level determined by the density of the nodes and the beaconing frequency. When receiving such a one-hop broadcast nodes do not react on it with a new transmission directly. They instead integrate the content of the message into their local knowledge base. With the next beacon that they will send this information or a part thereof may then be further distributed.

In order to maximize the information dissemination with the restriction of only few vehicles equipped and without causing congestion in the network Xu and Barth propose to adapt the rate of the beacons [XB06a]. With increasing distance and depending on the average velocity of vehicles along a road the frequency of beacons to be transmitted is adapted. Similarly, Fujiki et al. also present in [FKUH07] a mechanism that adapts the rate of the periodic beacons whereas Wischhof et al. in [WER+03a] recommend to take into account whether a different view of the traffic situation has arisen from the received message. Additionally, Wu et al. analyze the environment of a vehicle in [WFGH04]. If new neighboring vehicles appear, a beacon is sent to update their local knowledge base.

In [SFUH04] Saito et al. use a TDMA approach where one second is divided into ten slots. The used slot is dependent to the current speed of a node which influences the transmission interval of the periodic beacons. The idea is that if there is a traffic jam the current speed will be low and thus only few beacons need to be sent.

Torrent-Moreno et al. present in [TMSH05, TMSH06, TM07] an approach to control the channel load used by safety beacons and to provide a more reliable information transmission to receivers close to the sender. Instead of varying the frequency of beacons like in the above mentioned studies they propose to adjust the transmission power of beacons. The goal is to diminish the risk of packet collisions respectively congestion and thereby increasing the likelihood that beacons are successfully received.

Improving Network Connectivity

Oncoming traffic It has often been observed that making use of oncoming traffic for information transport brings large performance benefits. Especially when information about traffic or road conditions is exchanged, it is vital to communicate the data to vehicles driving far behind the region where the observations are made. Transportation by the locomotion of the creator of the information and by other vehicles driving in the same direction is thus not helpful. However, using oncoming traffic to forward the information can largely improve the performance of the system. For instance, Yang and Recker as well as Agarwal et al. show this effect in [YR05] and [ASL07, ASL08], respectively. Nadeem et al. even conclude in [NSI06] that it is efficient to use *only* cars going in the opposite direction to disseminate information in a fast way.

Roadside Units Yang and Recker aim in [YR05] to answer two questions on information propagation: what is the minimum necessary penetration ratio, and which other requirements need to be met in order to disseminate (warning) messages faster than by the pure locomotion of vehicles? The main focus here is whether it is possible to transmit data quickly to distant regions with only few vehicles equipped. They conclude that at least 20 % should use VANET technology. However, the results depend significantly on the communication range.

Independent from any specific penetration ratios or equipment densities, however, Yang and Recker make the—later often confirmed—observation that especially during rollout, when very few vehicles are equipped, extra measures ensuring connectivity to some extent will be necessary for many applications. They suggest to analyze the impact of additional infrastructure—road side units. Such infrastructure devices have appeared in many different flavors in the literature. In [GBMY97] Goodman et al., for instance, suggest that external information such as gas prices may be provided via "information kiosks", or simply by a WiFi access point. Other terms that have been used in the literature include "(stationary) supporting units" and "roadside units (RSUs)". Here, we will use the latter term.

RSUs may generally be either stand-alone devices that communicate only with vehicles via wireless communication, or they may be interconnected via a backbone network, which in turn could be realized either by wired or infrastructure wireless (GSM, UMTS,…), or via a mesh network of the RSUs. Essentially, their purpose is always to connect the VANET to external information sources, or to increase connectivity and/or capacity of the network. In [ZZC07] Zhao et al., for instance, propose to use supporting units at road junctions. When installed at traffic lights these devices are able to receive information by passing by cars and will inform newly approaching vehicles with this data.

The idea to use this kind of infrastructure devices is not new. Back in 1964, Covault and Bowes presented in [Bow64, CB64] results from a testbed using roadside radio communications. They analyzed if it is feasible to install RSUs in order to provide the drivers with additional information and to exert traffic control. Besides the technical challenges of such a system, user acceptance and cost are questioned. Although the technical prerequisites were quite different the goals of the roadside units were comparable to the ones discussed today. Different incidents were tested that had some influence on safety or traffic flow. Radio devices installed in the cars received the messages from the roadside units when passing by. In contrast to digital data messages used today, they used spoken messages output via a loudspeaker.

In [BCTL08] Banerjee et al. theoretically analyze the performance of different kinds of supporting units. In particular, they consider backbone-interconnected RSUs or base stations, a mesh network of RSUs, and stand-alone RSUs. They compare the number of stations necessary to keep the packet delay low. One conclusion is that a lot (5–7 times) more stand-alone RSUs are needed in order to achieve the same performance as with interconnected base stations. Banerjee et al. use an abstract, random node movement model. For stand-alone RSUs, node movement is important to deliver information to more distant regions. In real VANETs, nodes often move preferentially in one direction (e. g., towards the city center during morning rush hour). As mentioned above, typical long-range dissemination applications require that information travels *against* the main movement direction. Interconnected RSUs can meet these requirements and quickly deliver information to distant regions—much more effectively than stand-alone RSUs can. Therefore, in real VANETs, the performance benefit of networked RSUs can be expected to be even larger.

The concept of RSUs is of course not restricted to being used with beaconing-based dissemination schemes. In [CVCHO07], for example, Chennikara-Varghese et al. analyze how stand-alone or interconnected RSUs provide easier communication between clusters of vehicles. Here, RSUs are basically access points placed at the road side, which participate in the clustering protocol.

Algorithms for the Decision on What to Transport

While discussing how to transport information within the network, in many studies it is also considered which parts of information or which messages should be transmitted. In many studies relevance functions are used to order the available information with respect to its novelty or importance. Caliskan et al. use in [CGM06] the age and the distance to a resource for their relevance function. In [DJ07] Dornbush and Joshi classify the received messages according to their “significance”. When transmitting local knowledge to other vehicles, a message is filled until the maximum packet size is reached. This keeps the message size at a fixed level and avoids network congestion. Other metrics to decide which part of the local information should be transmitted are proposed as well. In the study of Saito et al. the authors suggest to use different approaches [STUH07]. In a probabilistic selection method no ordering of the information is performed but the data to be sent is determined in a purely random fashion. They also propose to order the information based on its age. Where other authors only make use of the newest and thus most current information, in this study it is also considered to propagate older information in order not to loose it.

Fujiki et al. assume in [FKUH07] that information about the vicinity of a vehicle is required more often than about distant region. It is hence useful to transmit these data more frequently. In [AEK+06, AESS06] Adler et al. suggest to compute a global message benefit of a message to be sent. This benefit is used to parameterize the medium access procedure. Since a highly relevant message should be sent first, the contention phase is modified in order to prioritize the message. This is performed by adapting the size of the contention window. However, in an environment with many transmissions and considering a more realistic (probabilistic) radio propagation model Torrent-Moreno et al. showed in [TMJH04] that such a prioritization method would most likely not achieve the desired effects.

An implicit or soft-state approach to share information only in a bounded area is presented in [XOW04]. Xu et al. assume that if two vehicles meet they will exchange their local knowledge. In order to limit the size of the information only the newest entries are transmitted. On the other hand, by adding new information to the local knowledge base, existing data will age and die out after some time.

Apart from being able to map position information to logical entities in the road network, the (static) parameters of the road network known by a navigation system (like, e. g., typical speeds on individual roads) may also be of great use when generating data that is helpful in VANET applications. It is a design option to inform other vehicles only in case of significant deviations from “typical” values. For example, a traffic information application might report travel times only if they are significantly different from what would usually be expected on the respective road. This has, for example, been proposed in [GIO04] and in [DJ07] by Goel et al. as well as Dornbush et al., respectively. Such an approach can potentially save significant bandwidth. In order to judge locally whether such a deviation currently exists, the respective travel times from the navigation system’s static map data may serve as a reference.

However, there is a general problem with such “minimalist” approaches, which transmit information only under specific circumstances: in case no information is received, it is not clear whether this is due to the fact that the situation is “normal”, or if there are simply no current measurements, so that nothing is known. Again, depending on the way the information is used and interpreted by the application, this may be acceptable or not.

2.3.3 Summarizing Measurements

So far, we looked at how local measurement data can be obtained, and we saw quite a number of approaches how this information can be transported to interested receivers. However, distributing all individual measurements made by all cars to all recipients who are interested in the respective road or area may be theoretically conceivable—but it is evident that this cannot scale. Therefore, mechanisms are required that reduce the amount of data dynamically and within the network—while the environmental parameters monitored by the system vary, while the positions of the cars and network topology continuously change, and while data is being exchanged wherever opportunities arise. In the remaining part of this chapter, we thus now focus on the problems of *summarizing* and *aggregating* measurement data.

In VANET dissemination applications, it is often the case that multiple measurements on the same or on closely related observed entities are available in the network. For instance, multiple vehicles may have traversed the same road segment, each of them measuring its own individual travel time. Or multiple vehicles may have obtained information on the current occupancy level of a parking site, but they may have done so at slightly different points in time. When such data—referring to the same entity, but to different observations—"meet" in the network, capacity limitations suggest that they are "merged" in some way. If they are combined into one single value, it is not necessary to store two individual values locally. Much more importantly, however, it is then also not necessary to transmit two values when information is exchanged between network participants. The aim of the combination operation is to generate one single data item which reflects the known information as good as possible. We call this process of combining multiple observations *summarization*.

A close relative of summarization is *aggregation*, and the distinction between those two is not always perfectly clear in the literature. In our terminology, aggregation means that data concerning *different* entities are combined into one value. The intention of aggregation is to produce coarser data representations of areas in larger distances, which can save even more bandwidth than combining multiple measurements of the same parameter through summarization. We will discuss aggregation in the next section.

Blind averaging In the simplest form of the data summarization problem, a device participating in the VANET is confronted with two distinct values for one and the same parameter (i. e., for the number of free parking places on a specific parking site, or for the time needed to travel along some specific road segment): one of the value may be stored in its local knowledge base, another value is obtained from some information source.

For example, the car may have made an observation of the parameter itself and it may have received the respective value from another car.

Unless additional information like, for instance, a timestamp is stored and communicated along with the data, it is not straightforward to decide if the current value in the knowledge base should be updated or not. In [XB06b] Xu and Barth discuss the problem of data summarization in the context of measuring travel times along road segments. They establish fixed ten minute time slots, and summarize the travel time measurements on the same road segment falling into the same time slot. In the first approach they suggest, when the travel time along a road segment is received from another car, the received and local values are averaged, and the average value is stored in the knowledge base. However, it turns out that this approach does not perform particularly well. The problem is that measurements made early during a time slot will dominate the dynamically updated average. When additional measurements are made later on, the early values are already known to many cars, so that they are often received. Therefore, the new measurements are quickly canceled out through repetitive averaging with the old values.

Timestamp-based comparison This problem motivates to complement the measurement values by timestamps. When a car makes an observation, it stores not only the observed value in its knowledge base, but also the current time. When the data is transmitted to other network participants, the timestamps are sent along with the values. A straightforward "summarization" mechanism—actually one that is often used in the literature—is to replace older measurements by newer ones, based on comparing the timestamps. Such an approach assumes that contributing entities have sufficiently accurately synchronized clocks available. This is not unreasonable since VANET-enabled cars will typically use some geopositioning system like GPS anyway. These systems can serve as a highly accurate time source, and therefore the requirement of synchronized clocks is not too severe a constraint.

A timestamp-based comparison mechanism certainly avoids that old measurements persist in the network and dominate newer, more up-to-date data. However, it also means that the value in the knowledge base of each car will always stem from one single observation. Whether this is a problem or not depends on the application; in particular it depends on the specific data source: in a system like the one proposed by Caliskan et al. in [CGM06], where fixed infrastructure (here, at parking sites) generates virtually exact measurements, newer values will indeed always describe the current situation better than older ones. Thus, in this case, replacing old measurements based on newer ones is certainly appropriate, and is an easy way to achieve good data quality.

In other situations, however—especially when the measurements are noisy—keeping only the most up-to-date value is not a good idea. When measuring link travel times through direct observations by the cars, influencing factors like individual driving style, traffic light phases, etc., may cause large variations. One individual measurement may therefore not reflect the actual situation very well.

Timestamp-based averaging Xu and Barth propose in [XB06b] two improved alternatives to the blind averaging scheme discussed above, in which they obviate the averages being dominated by old, but widespread measurements on the one hand, but also avoid to discard old data completely upon availability of new observations. Both algorithms assign timestamps to values, essentially in the way discussed above. The first approach is based on assigning the older value a lower weight upon averaging: when a new observation is made or when a value is received from another car (regardless whether its timestamp is newer or older), a weighted average is calculated, where the older of the two values is assigned a lower weight than the newer one. Xu and Barth found that good results are achieved when the two weights are chosen as 0.8 and 0.2.

As their third approach, they propose to update the values in the local knowledge base only when the car itself makes a newer observation. Other cars of course still assist in distributing the averaged values. In their scheme, vehicles store the number of samples in the current time slot so far, denoted by n, the timestamp T of the latest observation that contributed to the value, and the average travel time value v itself. A newly observed value o of the parameter can then—in the observing car itself—be integrated by an appropriately weighted average, generating the new value v', in the following way:

$$v' = \frac{1}{n+1}\,(nv + o)\,.$$

When such an update is performed, T is updated to the current time, and n is increased by one. The parameters n and T are communicated to other cars along with the current average value. When data is received from other vehicles, based on the timestamp T either the local value or the received value is kept; the older of the two is discarded.

This scheme ensures that each sample is included only once in the summarized values, and that all the measurements contributing to the average value have the same weight. It may still happen, though, that observations made by a car and locally included in the value are lost in later steps. This occurs in the following situation: assume that a car A makes an observation and includes it into its local knowledge base. In the course of this operation it also updates the associated timestamp. A short time later, car B has not yet received the data updated by A, but itself makes an observation of the parameter, and integrates it into its respective local value. The data in B's local knowledge base

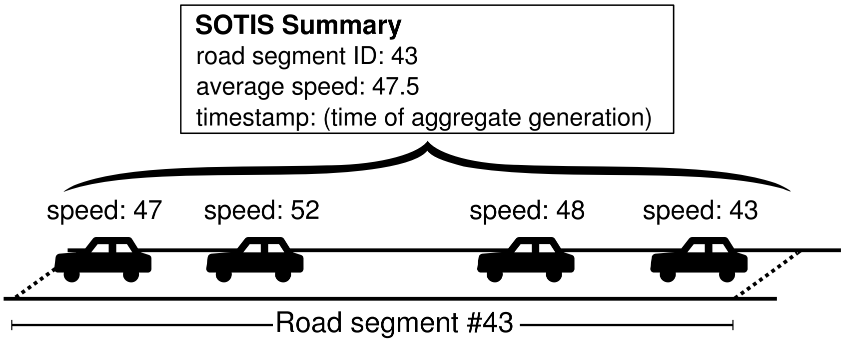


Figure 2.3: Per-segment vehicle velocity summarization in SOTIS.

does not contain the observation made by *A*, but it bears a newer timestamp. When these two data items are subsequently exchanged in the network and compared for up-to-dateness, *B*'s data will "win" and replace *A*'s data, thereby overriding the update operation made earlier by *A*. Consequently, *A*'s contribution will be lost. Despite this effect, Xu and Barth find that out of the three schemes they propose, this approach yields the most accurate representations of the exact travel time values.

Road-segment averaging A different variant of an averaging scheme is used in SOTIS [WER05] by Wischhof et al. The general mechanism applied there for summarizing information from multiple sources is called Segment-Oriented Data Abstraction and Dissemination (SODAD). SODAD is based on a subdivision of longer roads into segments. The road segments are hard-coded, so that all vehicles agree on the same set of segments, and each segment can be addressed with a unique ID. Locally, individual measurements are exchanged between the vehicles, so that a typical vehicle will be able to collect multiple samples from its immediate vicinity. For disseminating information to further away network areas, vehicles combine the samples they have for the road segment they are currently driving on. Upon generation, the per-segment values are assigned a timestamp—the time when the summarization has been performed. When they are disseminated in the network, the timestamp determines whether the value currently in the knowledge base or a newly received value are kept.

Wischhof et al. state that the function determining how exactly the individual samples are combined into a per-segment summary is application-dependent. In the concrete case of SOTIS, the application is based on exchanging information on the velocity of vehicles. The measured parameter is therefore the current driving speed. For summarizing the velocity information of multiple vehicles into a per-segment value, the average value is calculated. This is schematically shown in Figure 2.3.

A concept exhibiting some parallels with SODAD is outlined by Brønsted et al. in [BHK05, BK06] in the LIWAS context [LIW]. The Zone Diffusion Protocol described there also divides the roads into so-called "cells", and uses an application-specific data combination policy to merge observations made on the same road cell. As an example application the paper considers a road condition dissemination application, and suggests to use a conservative estimation policy: in case the information on the road condition for a cell diverges, the worst case is assumed, i. e., if one measurement classifies the road as icy will some other car considers it dry, it is conservatively assumed to be icy.

2.3.4 Geographical Data Aggregation

The summarization techniques discussed in the previous section allow to compare and merge data from multiple observations concerning the same parameter, like the current driving velocity on the same road segment in SOTIS or the number of free parking places on the same parking place in the parking guidance system by Caliskan et al. in [CGM06]. Thus, summarization allows to keep the amount of data per measured parameter (parking lot,...) constant. In larger road networks—for instance in a whole city or a network of highways—, there will, however, still be a huge amount of data left: the number of road segments in a city, for example, is high, as is the number of parking places in a city. Generally, if observed parameters are homogeneously distributed over the area of the network, then their number will increase quadratically with the covered radius. The limited bandwidth available for dissemination applications in VANETs will therefore not allow to continuously spread such detailed information over long distances, always maintaining a fine-grained picture in the whole network.

To overcome this limitation, it has been proposed to use in-network data aggregation techniques, which aim to combine the current values of multiple parameters into one single aggregate value. Such an aggregated value could be the average speed on a longer part of a highway, or the total number of free parking places in a larger part of a city. In this section, we will discuss data aggregation schemes for VANETs.

Clustering groups of similar vehicles The TrafficView [NDLI04a, NDLI04b, DND+04] system by Nadeem et al. aims to facilitate distributed cooperative traffic monitoring of the highway ahead of the vehicle. While this goal is similar to that of SOTIS, TrafficView manages data on the current position and speed of individual vehicles rather than on averages for road segments. Because the amount of this data soon exceeds reasonable limits, an adaptive aggregation mechanism is used. Locally each car stores a list of vehicle IDs, positions, and speeds. Upon transmitting information to neighboring vehicles, the size of the transmission is limited, and therefore mechanisms are introduced to reduce

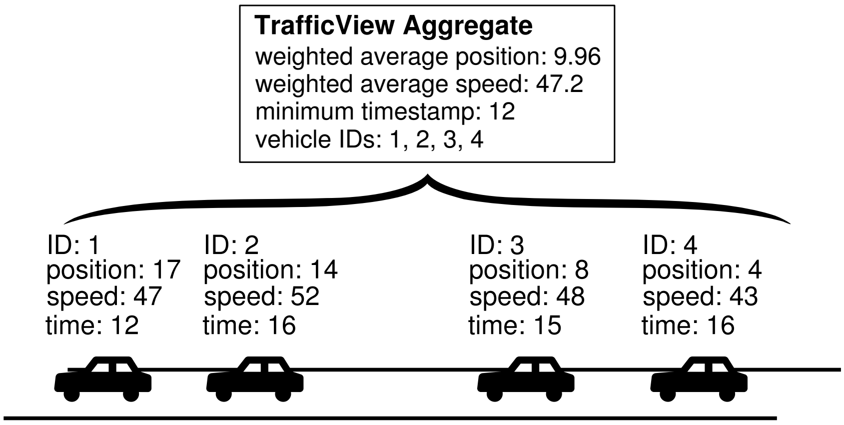


Figure 2.4: An aggregate over four vehicles in TrafficView; the calculation of the weighted averages here assumes that the aggregating vehicle is at position 0.

the amount of data while at the same time conveying approximate information about as many vehicles as possible. In their transmissions, TrafficView nodes therefore use aggregate records. Aggregate records describe not single vehicles, but groups of vehicles with similar properties.

An aggregate record in TrafficView consists of one speed, position, and timestamp value, along with a list of vehicle IDs. This saves bandwidth for transmitting separate values for each individual vehicle. The speed and position values in the aggregate record are weighted averages over the respective parameters of the vehicles in the record. An aggregate is schematically shown in Figure 2.4. TrafficView proposes aggregation algorithms that are parameterized for regions of the road ahead. The most important input parameters are an aggregation ratio, indicating how many vehicles should be combined into one aggregate, and the amount of data that may be transmitted for a given region. Based on these parameters TrafficView then aggregates appropriate values.

The CASCADE scheme [IW08a, IW07] proposed by Ibrahim and Weigle shows some similarities to TrafficView's aggregation approach. It also distributes information on individual vehicle positions and velocities with the aim to support dynamic route planning by networked navigation systems. CASCADE uses syntactic compression on clusters of similar vehicles, where similarities again means similar position and speed. Groups of vehicles are summarized in one record, where the amount of per-vehicle data within the record is minimized.

CASCADE vehicles have detailed information about individual vehicles in their neighborhood, so called "primary records". The entirety of the detailed knowledge of a vehicle is termed "local view". The vehicles on which information is available are grouped

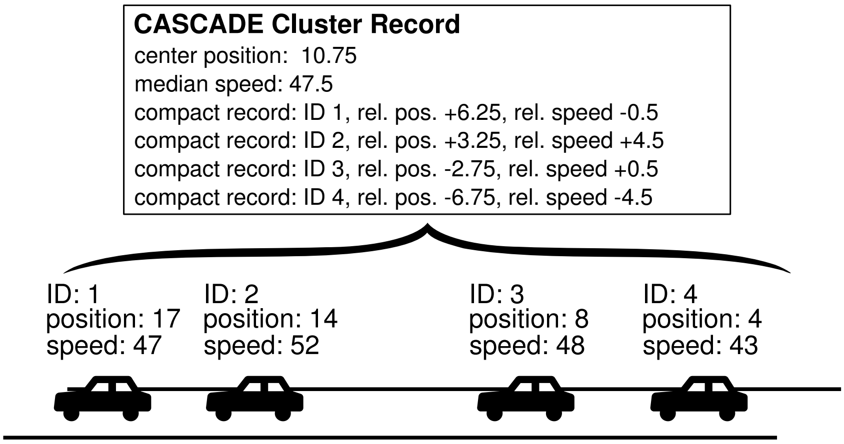


Figure 2.5: An aggregated cluster record in CASCADE.

into clusters depending on the distance from the local vehicle, their heading, and their altitude. The per-vehicle data in the cluster records is then compressed by expressing vehicle coordinates and speed relatively to the cluster center and the cluster's median speed. Consequently, an aggregated cluster record contains the (absolute) cluster position and speed, and one "compact record" (with coordinates relative to the cluster) per vehicle in the cluster, as it is shown in Figure 2.5. Aggregated information is distributed over longer distances. When aggregates records are received, they are incorporated into the "extended view" of the receiving vehicle.

SOTIS, TrafficView, and CASCADE share a number of similarities. All three systems collect information on position and speed of vehicles, and all of them combine information on vehicles with similar parameter values. None of these systems supports hierarchical aggregation of information on a whole road network, i. e., information on distinct roads is never combined. Thus, they are borderline cases between summarization and aggregation. Here, we decided to classify SOTIS' mechanisms for data reduction as summarization, and TrafficView's and CASCADE's as aggregation, because SOTIS considers vehicle data as samples of the same parameter—the velocity on a road segment—, and then summarizes these samples. In TrafficView and CASCADE, in contrast distinct vehicles are consequently treated as distinct observed entities.

Distributed data clustering In their StreetSmart system [DJ07], Dornbush and Joshi tackle the problem of a distributed traffic monitoring system from a different angle. In StreetSmart, each vehicle records samples—essentially position and speed—along its own movement path. Individual position data in StreetSmart is represented by a road ID and an offset (i. e., position) along the road.

These samples are not exchanged between vehicles; they are locally aggregated using a data clustering mechanism with an application-tailored similarity measure. This measure depends on *i)* the road the measurement was made on, *ii)* the position along the road, and *iii)* the current speed. It is used to identify groups of samples which exhibit unexpectedly low speeds on the one hand, and are otherwise similar in terms of the above mentioned criteria. If clusters exhibit a speed which is not typical for the respective road, traffic congestion is assumed to exist. Clusters for which this applies are then communicated to other network participants, abstractly described through their cluster centroids.

So, while StreetSmart also uses clustering of similar data sets, these data sets are position/speed samples collected over time from the *same* vehicle, while the previously discussed systems TrafficView and CASCADE aggregate position/speed records from different vehicles.

Aggregation over hierarchical areas The approaches discussed so far aggregate data from multiple vehicles, or multiple measurements consecutively made by the same vehicle. They are therefore able to significantly reduce the amount of data to be exchanged. However, they alone are not sufficient to fully overcome the scalability problem, for two reasons. First, they can sensibly combine only data from the same road, driving in the same direction. So, the total amount of data in the network still quickly increases with the size of the covered road network.

The second and probably even more important reason is that the amount of data for a complete picture still increases with the number of vehicles: in TrafficView and CASCADE, the aggregates contain lists of the IDs of included vehicles. So, there is a reduction of the number of bytes used per vehicle, but the size of the aggregates still grows linearly with the number of contained data items. In StreetSmart, local measurements are clustered, but each car distributes its centroids in the complete network, without further aggregation; therefore, also there, the amount of data increases with a larger number of vehicles (or, equivalently, a larger covered area). We will thus now look at proposals where the size and count of aggregated data representations is independent from the number of measuring vehicles, and where the amount of data does not increase linearly with the covered area (or with the number of road in the covered area).

The first approaches which achieve such a behavior have been proposed by Caliskan et al. in [CGM06]. In this work, area-based aggregation models are used: with increasing distance, measurements from larger and larger areas are combined into aggregated values. The decentralized parking guidance scheme described there uses such an aggregation scheme. Local infrastructure at each parking site is used to locally summarize

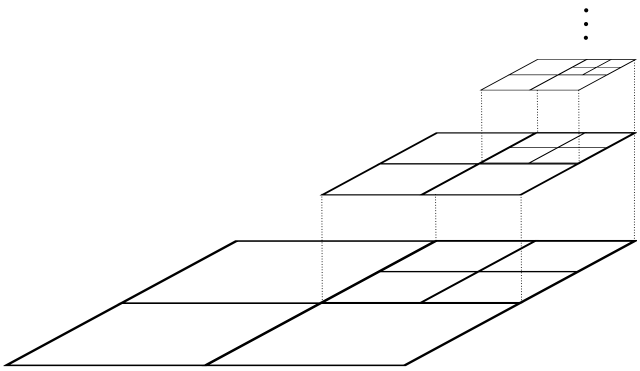


Figure 2.6: A quadtree.

information on the overall occupancy of the parking site. This is founded on the assumption that larger parking sites will typically have infrastructure for payment and/or admission installed anyway. Such infrastructure already has an up-to-date overview of the occupancy level of the site, so potentially noisy measurements conducted by the vehicles themselves are unnecessary.

The "atomic" information on the current occupancy status of a single parking site is handed over to passing-by cars, and is subsequently passed on between vehicles. Within the network, aggregates are formed. They serve as a compact representation of the parking situation in larger parts of a city, by describing the total number of free parking places on all parking sites within some larger area. A fixed subdivision of the city is used, defined by a quadtree over the two-dimensional plane. A quadtree is a hierarchical subdivision of the plane into smaller and smaller squares, as indicated in Figure 2.6. Within a certain radius, information on all individual parking sites is exchanged in the network; at larger distances, only aggregate values are used. These aggregate values become coarser and coarser the more distant the described area is. The motivation for hierarchical aggregates is clear from the application focus: at a larger distance, it is sufficient to know about the approximate situation and the coarse direction where parking places can be found. It is not necessary to know the detailed occupancy level of all parking sites. The closer a car gets to its destination area, the more fine-grained the information gets, so that a decision for one specific parking site can finally be made.

Vehicles form aggregates of the lowest level (i. e., the cells in the smallest quadtree grid) by adding up the number of free parking places known in the area. That is, if a vehicle has received atomic information from a number of parking places in the same quadtree cell, it may generate a lowest-level aggregate. Aggregates for usage at again larger distances use the higher levels of the quadtree hierarchy. They are generated from aggregates of the next-lower level, by adding the respective numbers of free parking places from the four component sub-aggregates. So, larger aggregates are hierarchically created from smaller aggregates.

An aggregate essentially comprises the ID and hierarchy level of the respective quadtree cell, a timestamp, and the total number of free parking places. Thus, each aggregate has the same size, regardless of the size of the area it covers and of the number of parking sites within that area. Consequently, for an increasing aggregation level—and thus with increasing geographical distance from the described area—the amount of data per covered area decreases.

Comparing and merging hierarchical area aggregates In the previous section we saw that by using timestamps it is easily possible to keep only the most up-to-date value observed for some parameter, discarding older measurements. This is possible because the up-to-dateness can be compared based on the timestamps. Unfortunately, this is not possible for aggregates describing more than one parameter. Observe that an aggregate generator—a car in the network—will typically not have "perfect" (i. e., complete and fully up-to-date) information available for the generation of the aggregates. Typically, it is not even required that a car has complete coverage of the underlying information; aggregates may even be formed when parts of the underlying data is missing. Therefore, aggregates will also not be "perfect" representations of the current situation in the area they cover, but only the best possible approximation based on the current knowledge of the generating node. Thus, if different nodes generate aggregates for the same area, these aggregates will typically be based on different, but partially overlapping information. For example, one node may have more up-to-date knowledge about one part of the underlying data, and another node's information is better with respect to some other sub-area. The fundamental issue that therefore arises is that the completeness and up-to-dateness of aggregates can not be expressed through a single timestamp. A data structure describing the set of contained information and the respective timestamps is obviously also not an option, because it would compromise the aim of an aggregate size which does not grow with the amount of data contained.

Summarization and aggregation mechanisms deal differently with this problem, but typically they resort to using some kind of "best guess" heuristic timestamp. For example, SOTIS uses the time of aggregate generation as a timestamp for the aggregate; obviously, however, it is well possible that a node with old and incomplete information generates an aggregate, which is then considered more up-to-date than a slightly earlier generated aggregate with a more solid and (at least for the most part) more up-to-date basis. The aggregate timestamp generation schemes for TrafficView (where the timestamp of the oldest observation included) and Caliskan et al.'s parking guidance system (which calculates the average timestamp of the components used for aggregation) exhibit similar artifacts; there, too, situations are easily constructed where aggregates with quite comprehensive and up-to-date coverage are assigned an older timestamp than aggregates with a much weaker information basis.

The quadtree-based approach used in [CGM06] allows for an easy way of addressing areas and their hierarchy. But it has the drawback of not considering the topology of the underlying city: the same quadtree field may include areas which are barely connected in the road network, and the properties of which may largely differ—for instance, areas on two sides of a river or a railway. They argue that aggregation areas should follow the topology of the underlying road network, and give a generic formulation of an aggregation hierarchy, along with some hints on how it should be designed to allow for effective sketch-based aggregation.

Chapter 3

Simulator Interlinking for Car-to-Car Communication

We have shown in the previous chapter that the development and evaluation of applications for vehicle-to-vehicle communication plays an increasingly important role, both in scientific and industrial research. These applications aim at enhancing traffic safety, driving comfort, and in-car entertainment. The common development cycle for these applications oftentimes starts with a theoretic and abstract description of the problem. By splitting the problem into several sub-problems a refining process is performed. At the end of this procedure an implementation of a prototype will be achieved. However, in order to reveal hidden challenges or design flaws a detailed simulation evaluation step preceding the actual implementation is performed more often than not.

In order to cover the whole spectrum of parameters diverse studies are commonly conducted. Each study is focused on a specific detail of the implementation or process. To model this, various simulators are used. The simulators are built just for one special purpose which can be evaluated in great detail while abstracting from the reality in many cases. Conducting multiple simulation runs with different specialized simulators does, however, not allow a holistic evaluation. In the case of a VANET for instance it is thus not possible to take into account the exact transmission of data packets while moving vehicles in a realistic manner. Up to the present, many studies use movement patterns that are created by vehicular traffic simulators and are stored in traffic log files. In a second step a network simulator is used to model the algorithms employed for routing data packets through the VANET. It uses the traffic log files generated by the traffic simulator to "move" the nodes during the simulation. Realistic implementations of applications are also waived and replaced by some artificial stochastic or deterministic processes within the network simulator.

One key disadvantage of this approach is the inability to modify the vehicular traffic data in response to application layer events. No matter what happens at the application layer, the movement of a vehicle will remain the same. There is no way for an application to change the behavior of a vehicle during runtime, e. g., to reduce the velocity of a car or to make a turn at a junction. This significantly reduces the level of realism for key applications such as navigation systems which in reality do influence the movement of a vehicle.

So as to solve this problem we propose to interlink different simulators for network simulation, traffic simulation, and application simulation. This will establish an integrated simulation environment or meta-simulator that goes beyond existing simulator coupling approaches like [HA04] or simple downstream simulators like [XMKS04]. This environment allows to do holistic evaluations of VANET applications or protocols. In order to develop this meta-simulator, two main challenges need to be dealt with: first, how to communicate between the simulators without encountering major performance losses and second even more critical, how to ensure (simulation) time synchronization between all participating simulators.

3.1 Architecture

The aim of our simulator coupling is to enable a holistic evaluation of VANETs with the ability to tune each 'adjusting screw' of all involved simulators. As depicted in Figure 3.1, the architecture comprises one simulator each for vehicular movements (VISSIM), application behavior (MATLAB/Simulink), and network functionality (ns-2). Although the architecture is presented in accordance with the aforementioned tools, it is not limited to these simulators and can be enhanced by several other tools. A helper application (Simulation Control) is present to encapsulate control messages and enables cross operating system interaction between simulators. In a setup Tcl-script the parameters like IP-address of the Simulation Control, the name of the scenario plus the usual parameters to conduct a network simulation like simulation time are set.

To avoid immoderate overhead the central element of the architecture is built by the network simulator. It controls the interlinking process and handles the timing of the simulation. At startup the network simulator is initialized with the parameters of the simulation (e. g., algorithms to be used, complete simulation time, etc.). As a first task the network simulator contacts the traffic simulator via the Simulation Control by sending the name and setup of the scenario. The traffic simulator then opens a corresponding scenario file and returns the geographical dimensions of the scenario. After the initialization process the network simulation starts. At the same time the traffic simulator starts its simulation as well and performs a single simulation step. The traffic simulator is then asked periodically for movement data of each simulated node. At any time it is ensured that the network simulator is able to modify the behavior of any node by sending a message to the traffic simulator.

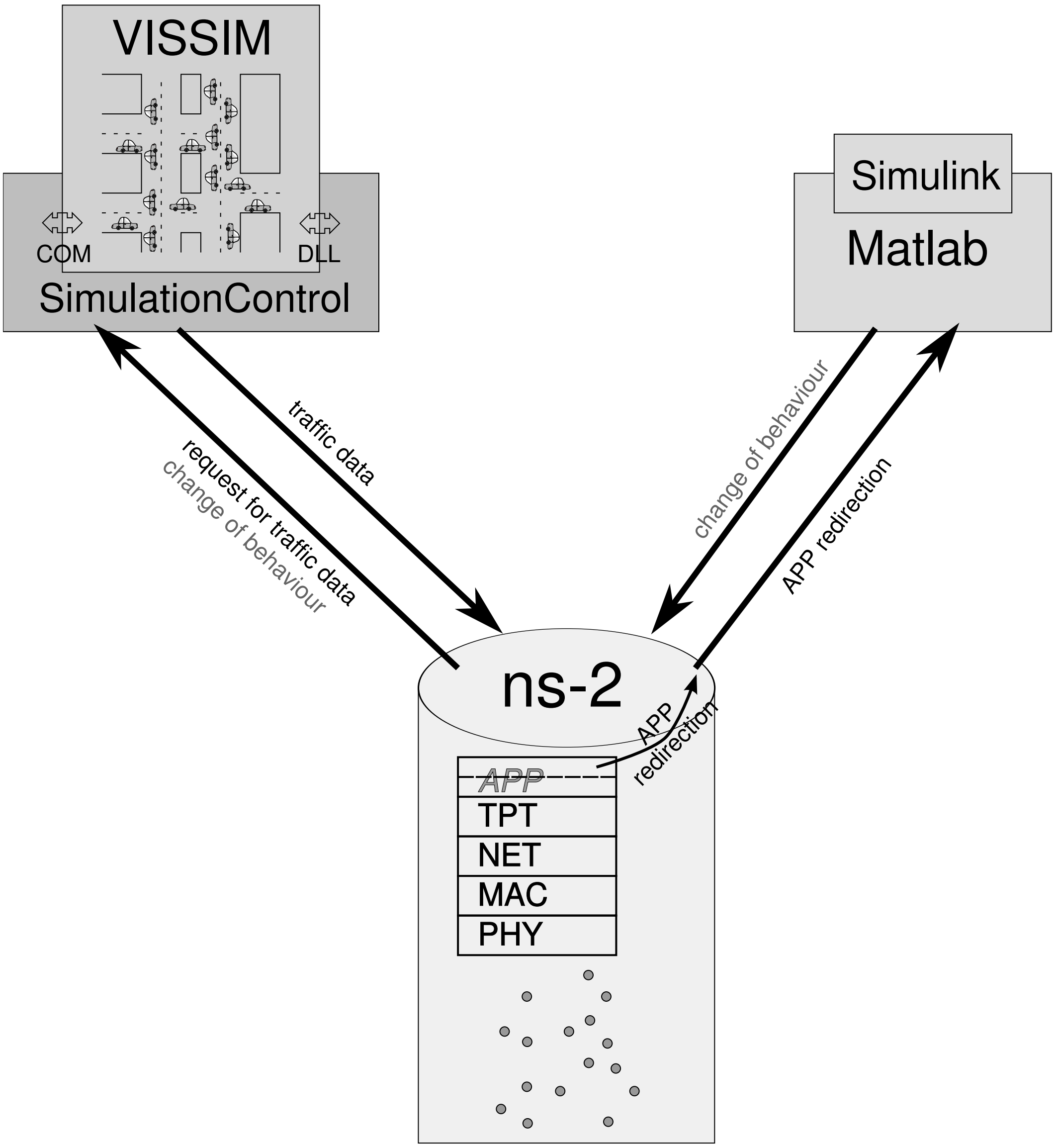


Figure 3.1: Schematical representation of the simulation environment.

Supplementary to the network and traffic simulators we integrate a simulator for application specific functionality. One reason for this is that many vehicle manufacturers use simulators to evaluate the functionality of applications for vehicles. Our architecture allows this simulation to serve both as a source of data packets and for determining when a driver changes her behavior (e. g., due to an emergency warning). Data packets that have reached their destination are handed to the application simulator and can then be processed by the simulated application. The application decides how to react on this received packet: it may modify the behavior of a driver or generate new data packets. Both information is handed to the network simulator again. Information on driving behavior is forwarded to the traffic simulator while additional data packets are handled by the network simulator itself.

3.1.1 The Network Simulator – Central Module

In the research community multiple distinct network simulators are employed. We decided to use the freely available and widely used open-source simulator ns-2 [ns2]. As stated above the architecture of the coupling allows to use other network simulators as well.

To enable the simulator interlinking, a wrapper class called *Synchronize* was implemented for ns-2. This class is responsible for the synchronization between all interlinked simulators. At the startup of the simulation a Tcl-object of this class is created and the connection to the traffic simulator is established. After this initialization phase the network simulator itself is instantiated. In order to adapt the simulator coupling to the discrete event simulator ns-2, control events are scheduled that trigger the interaction with the other simulators. After the expiration of an event a request for new movement data is sent to the traffic simulator. This process is periodically repeated. Depending on the application the time granularity between two movement requests is important. It can thus be set and changed dynamically.

3.1.2 The Traffic Simulator

Specialized traffic simulators facilitate the analysis of vehicle movement notably. The most common tools for the simulation of traffic use a distinct mobility model in order to map the reality as precise as possible. The VISSIM simulator by PTV AG [PTV] is based on the Wiedemann car-following model [Wie74] and is also able to simulate pedestrian dynamics based on the Social Force Model [HM95]. It includes, for example, multi-lane traffic, traffic lights, and different types of vehicles. It also takes realistic driver specific behavior into account. Especially in the U. S., CORridor SIMulation (CORSIM) [OZRM00, COR] is a widespread tool for the microscopic simulation of vehicles in the ITS context. In contrast to VISSIM and CORSIM, the Simulation of Urban MObility (SUMO) [KHRW02, SUM] is a very recent simulation tool based on an open-source philosophy. It is based on the Gipps-Family car-following model [KWG97].

For the simulator coupling we decided to use the VISSIM simulator because of its multiple interfaces. In order to interact with the simulator two different interfaces are offered: *i)* the Component Object Model (COM) interface [GHL+98] and *ii)* the customizable Dynamic Link Library (DLL) called *drivermodel.dll*. VISSIM requires a Microsoft Windows platform. Figure 3.2 depicts a screenshot of the graphical user interface.

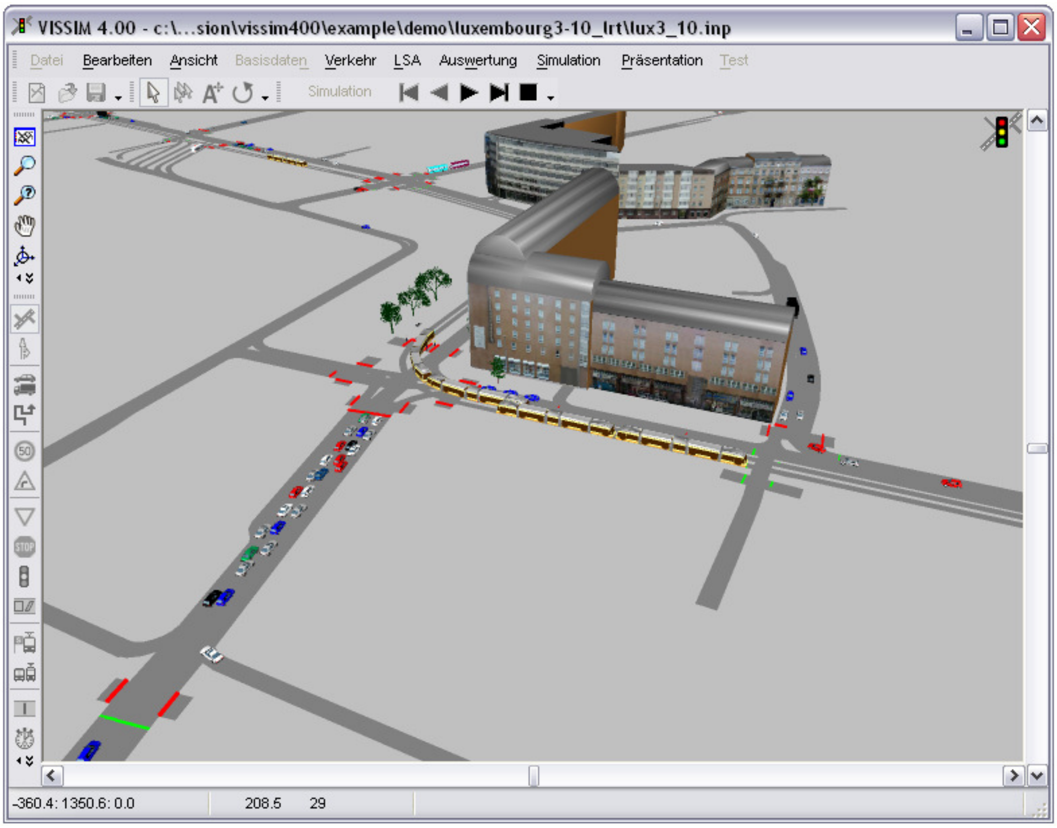


Figure 3.2: Screenshot of the main window of the traffic simulator VISSIM.

The COM interface allows full control about most aspects of a VISSIM simulation, including the modification of attributes such as speed and acceleration of a vehicle. Since the access to this interface is very heavy-weighted we use it only for those aspects that cannot be controlled via other means. This includes starting and halting the simulation process. In contrast to the COM interface, the Dynamic Link Library offers a very light-weighted access to the internals of the VISSIM simulator. This interface is originally intended to allow the customization of the driver behavior during runtime. In particular, it can be used to change the behavior of the driver in response to an application-level event. Furthermore, we employ the *drivermodel.dll* interface to access data such as the positions of the vehicles in a lightweight and efficient fashion.

Simulation Control

Due to the fact that VISSIM requires a Microsoft Windows platform while the other simulators perform optimally under Unix operating systems, cross operating system communication is necessary. This task is performed by the external application *Simulation Control*. It acts as a wrapper application for any traffic simulator allowing to provide a uniform interface for the access.

At the initialization phase the Simulation Control is contacted by the network simulator and a TCP connection is established. During simulation it receives the queries from the network simulator and converts them transparently either into COM commands or into commands for the *drivermodel.dll*. The communication between the simulation control and the drivermodel is realized using threads. In order to exchange data in a light-weight fashion we accordingly implemented a shared memory space.

For the external communication a sequence of structures is sent over the socket to the network simulator. Figure 3.3 depicts the packet structure as well as the data structure within a packet. These structures are used to update the movement of the nodes within the network simulator. They contain the new positions of the nodes. Due to different coordinate systems first a transformation to the coordinate system of the network simulator is performed. The last parameter is the speed parameter. It tells the network simulator how fast (in meters per second) the nodes are moving.

3.1.3 The Application Simulator

We decided to use the MATLAB/Simulink [Mat] environment as an application level simulator. MATLAB/Simulink is used by vehicle manufacturers to evaluate in-car applications. Simulink allows a developer to create applications by providing personally adaptable drag and drop utilities. Furthermore, it is possible to automatically generate C-code out of the Simulink simulation. This is regularly done to use the same code-base for simulation and real-world products.

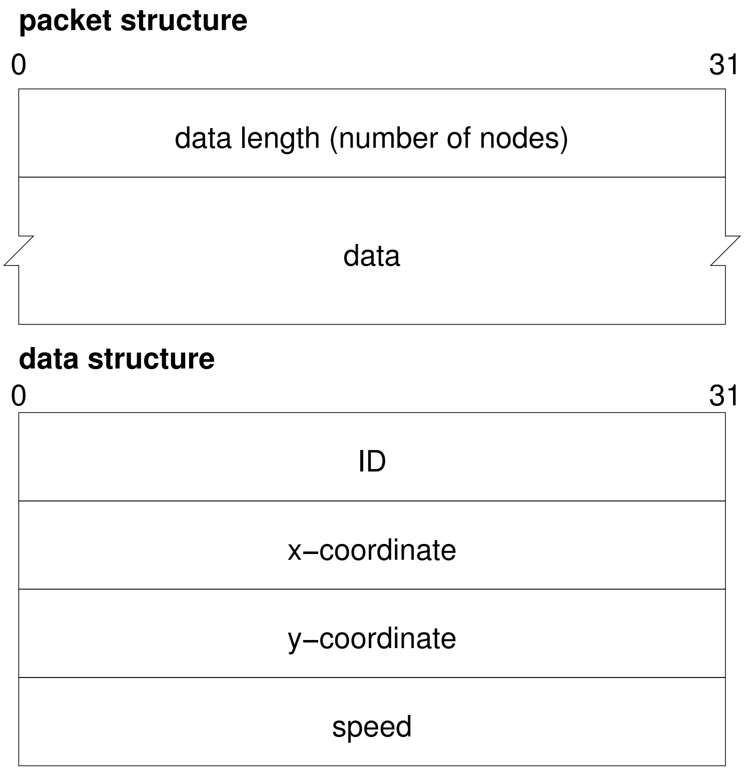


Figure 3.3: Packet structure for movement updates of nodes within the scenario.

The MATLAB environment is used to run the simulations. By default the MATLAB environment offers the possibility to operate by remote control through the MATLAB engine. This engine includes a variety of library functions such as starting and quitting MATLAB, exchanging data with MATLAB, or requesting mathematical operations. In our environment the network simulator uses these functions to communicate with MATLAB/Simulink during the runtime of a simulation.

The control of MATLAB is realized by a new controller class called *MatSimulator* within the network simulator. The setup is similar to the one of the *Synchronize* class. Within the Tcl-environment an object is instantiated and given to the *Synchronize* object for further control of the MATLAB program. By instantiating the MATLAB program, the simulation application is loaded and waits for messages from the network simulator.

There is one major performance issue with interlinking the network and the application simulator: handing data back and forth between the simulators is costly. One transfer from the network simulator to MATLAB takes approximately 20 ms. It is therefore very important to pool all packets for all vehicles that need to be delivered to the application simulator within a given period of time. These packets are de-pooled by the application simulator and are then forwarded to the simulation of the individual vehicles. From our experience it is not feasible to let one separate simulation register with the network simulator for each vehicle.

3.2 Obstacle Modeling

3.2.1 Simulation Aspects

In the evaluation process of VANET applications, the simulation of communication between cars plays an important role. In order to model the direct communicate between two cars, their position has to be taken into account. In highway scenarios nodes are mostly located in a one-dimensional fashion whereas in city scenarios nodes are located in a more two-dimensional fashion. So as to simulate the propagation of radio signals models have been developed that decide—depending on the distance between two nodes and the current noise level—if a packet can be received and decoded accordingly. This applies for deterministic as well as for probabilistic models. But especially in city scenarios not only the distribution of and the distance between nodes is important for a proper reception of packets. It is also very important to consider disruptive influences originating from the scenario itself. In the presence of buildings, for instance, the radio propagation can be interfered by walls that block radio signals. Nodes are thus not able to communicate with each other although they are close together. We will call this radio barrier an "obstacle" in the following sections. In order to determine the proper reception we present an approach to analyze the environment in a light-weight fashion.

Many approaches have been proposed that deal with obstacles. Most of them make use of ray tracing techniques to emulate the propagation of radio signals in a very realistic fashion. Examples have been presented for instance in [SMZ+02, Val03, LB98, KGJWI+99, CPSdAGB98, TT95, RWG97, IFTK00, SHR05, SJK+03]. To calculate the effects that happen while propagating through different construction materials it is, however, necessary to know about the specific structure of buildings and their effects on radio signals. If we assume that this information is available the results of the simulations, indeed, would be more realistic. The insights of the considered application are though only valid for this gauged simulation scenario. It is further apparent that the effort to model a whole city with ray tracing techniques will affect the VANET simulation procedure.

In the following paragraphs we will present an approach of an obstacle modeling that takes a different approach requiring only limited computation power. The mechanism is based on the line of sight criterion between two nodes. Only if there is a line of sight which is not interrupted by a building packets may be received.

The calculation of the line of sight in our implementation is a discrete decision process. It is based solely on shadowing effects. We do not consider effects like reflection, scattering, or diffraction of signals at these obstacles.

The simulation of this obstacle modeling is shaped by the results of multiple simulators. On the one hand radio propagation calculation is the main task of a network simulator. On the other hand the structure and locations of these obstacles are predetermined by the traffic model which is used by traffic simulator. It is thus necessary to enhance the simulator coupling as well as its components with the obstacle modeling. In the following paragraphs we will present two different approaches. Both gain similar results while differing in their functionality and input data.

3.2.2 Polygon Based Obstacles

A natural representation of obstacles in the simulators is given by polygons. To map a building into the considered scenario the contour may be drawn. Most traffic scenarios, however, do not model buildings and their placements. They only map the streets of a scenario. We decided therefore to add buildings into a scenario manually by implementing an auxiliary application for VISSIM. It is now possible to insert radio obstacles (buildings) into a considered scenario. We implemented this application such that it spans a transparent canvas on top of the VISSIM scenario screen. This canvas is synchronized with the underlying VISSIM screen in order to scroll or to zoom into a detailed view of the scenario accordingly. We can then draw polygons on this canvas that represent the obstacles. The drawn polygons will be saved in a file preventing a multiple (re-)construction of obstacles. Figure 3.4 depicts the obstacle input screen for an example city scenario. The standard VISSIM screen with the obstacle creation screen on top is visible.

So as to bind the VISSIM polygon obstacles to the ns-2 simulator we also extended ns-2 with an obstacle modeling. When starting a simulation run, ns-2 receives a list of polygons from the Simulation Control and stores this data in a local knowledge base. Without any obstacles, ns-2 would check if a sent packet can be received or rather detected by some other node based on the radio propagation model. The obstacle modeling is put on here. A new obstacle module determines the beeline between these two considered nodes. If there is a polygon representing an obstacle between these nodes the beeline will intersect with an edge of the polygon implying that no line of sight exists. It further implies that the packet would not have been seen *at all* by the target node. The simulation process will continue without regarding this "reception" at the considered target node. In contrast, if there is no polygon intersecting with the beeline it implies

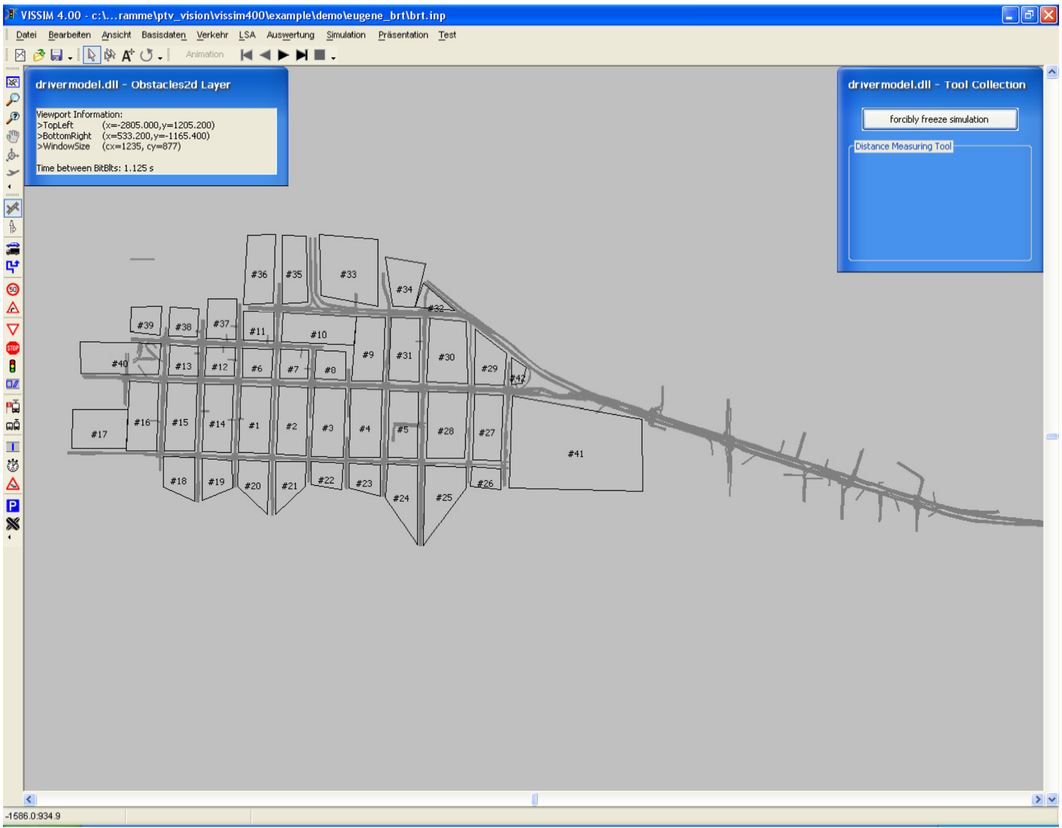


Figure 3.4: Part of a city scenario modeled with obstacles for the simulations. A user can draw obstacles represented by polygons on top of this VISSIM city scenario.

that there is a line of sight between these two nodes. The simulation process will be continued as if there was no obstacle modeling. This comprises for instance the calculation of the received signal strength and the following processes to handle this data packet.

3.2.3 Street Based Obstacles

Even though the manual construction of obstacles needs only to be done once for a given scenario, drawing polygons on the canvas of VISSIM scenarios by hand is quite ineffective, error-prone, and expensive. It is therefore desirable to automate this process. The idea behind our approach is to analyze a traffic scenario directly by parsing the scenario file. Ideally, the language which describes the scenario is based on a context-free grammar which is fortunately the case for VISSIM scenarios. These scenarios are written in a language which can be represented by a Backus-Naur Form (BNF) [Knu64]. It is thus relatively easy to efficiently parse these scenario files by making use of parser tools like *flex* [fle] and *bison* [bis].

In contrast to the procedure of static obstacle creation by polygons the street based approach utilizes the complementary structure by reading in the structure of the streets. In order to map the scenario to our model street segments are assigned a given width, either by their real width according to the scenario file or by a default width which is user definable. The obstacle model approach then assumes that everything which is not covered by a street (area) represents an obstacle for radio signals. As most commonly buildings are located at the side of streets and in order to deal with a worst case analysis we regard this assumption as valid.

Another advantage of automatically parsing a scenario file is that it can be performed directly by the network simulator. It is therefore not necessary to transmit this information between multiple applications, thus decreasing the communication overhead of the coupling.

Due to the different structures of these obstacles the obstacle modeling of ns-2 has to be adapted accordingly. The basic idea is similar to the polygon based obstacle modeling: if there is a line of sight (LoS) between two nodes, then direct communication between both may take place. We use an algorithm that imitates a step-by-step line of sight construction. Starting from a sending node it aims to find consecutive road segments in direction to the target receiver. The LoS exists if it is completely covered by street segments.

The algorithm to check this criterion proceeds as follows: first, it starts with the road segment the sender is currently on. Then a beeline between the sender and the possible receiver is created to limit the number of street segments that need to be considered. If the beeline leaves the area of the road segment at an intersection point, the algorithm tries to find an alternative road segment that contains this intersection. If there is no road segment abiding this condition the algorithm returns that there is no line of sight between these two nodes. The packet is not observed by the target node and the normal procedure of the simulator continues.

Figure 3.5 depicts examples from this algorithm. Node S sends a data package and the algorithm needs to calculate if there is a LoS between S and a possible receiver D. The beeline intersects the first road segment at point i_1. Another road segment is found which is once again intersected at point i_2. The last found street segment includes the target node thus returning a complete line of sight between these two nodes.

In contrast to this behavior, Figure 3.6 depicts an example of a LoS construction that fails. The first intersection point is calculated like in the example above. However, there is no further road segment that covers this intersection point while gaining any progress towards the target point D. The algorithm returns that there is an obstacle between these nodes impeding direct communication.

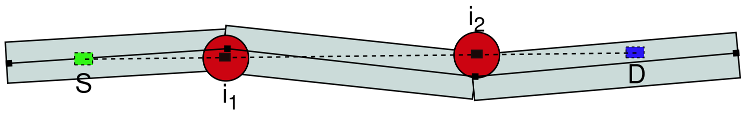


Figure 3.5: Line of sight construction.

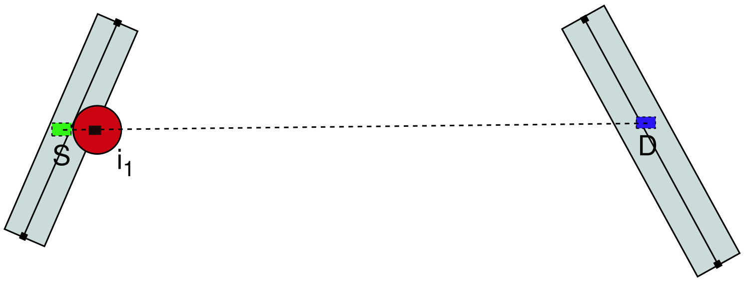


Figure 3.6: Line of sight construction failed.

3.3 Influencing the Movement of Vehicles

For the demonstration of the simulator coupling's ability we designed a simplistic VANET application. Vehicles are moving in a realistic scenario combining highway and city traffic as depicted in Figure 3.4. In the highway part of the scenario is vehicles drive on two lanes in each direction. In the city radio obstacles have been inserted with the polygon based mechanism. After the simulation of a traffic accident the affected vehicle stops at its current location. At the same time a warning message is created and sent to neighboring vehicles. Upon reception, the emergency warning is forwarded by using simple flooding techniques as mentioned in Section 2.3.2. Each vehicle that receives the message simply rebroadcasts it once and decelerates its speed using a regular braking pattern.

Multiple simulation runs were performed to get statistically relevant results. For each simulation run one vehicle near the center of the simulation area was selected to simulate the accident. In the network simulator an accident is scheduled and after expiring it triggers the stopping of the vehicle in the traffic simulator. The affected vehicle thus starts to transmit a 64-byte emergency warning message. One focus of this study is to analyze the speed regulation interaction between both simulators as well as to explore

Figure 3.7: Packet delivery ratio versus distance of communication partners.

how reliable and how fast the emergency message would be delivered depending on the available bandwidth and the distance to the accident.

In this study we do not use any application logic that is able to determine whether the vehicle receiving this warning message is actually affected by it (e. g., driving in the appropriate direction). For the experiments the average delivery ratio and delay of the warning messages are investigated. Vehicles are grouped according to their distance to the original sender of the emergency warning at the time this warning message is transmitted by the original sender. The first group contains all vehicles that are within a 500 m radius, the second includes all vehicles that are not in the first group but are located within a 1 000 m radius and so on. This value is displayed as distance on the x-axis of Figure 3.7.

The y-axis of Figure 3.7 shows the ratio between the number of nodes that have received a warning packet and the number of nodes that belonged to the respective group. From a reliability perspective it can be observed that a bandwidth of 10 KBit is the absolute minimum value to achieve a reliability of around 0.9 for any significant distances. It should be noted that this is influenced significantly by the topology of the radio obstacles. This seems also to be the reason why the group at 1 500 m and 2 000 m has a higher delivery ratio than the group at 1 000 m: in the topology of our city model distances between 1 500 m and 2 500 m required the traversal of a long street which is unlikely to fail.

Further conclusions can be drawn by studying the average latency of the first delivered warning packets depicted in Figure 3.8. This latency increases dramatically as the available bandwidth decreases. To transmit a warning message early enough to warn another

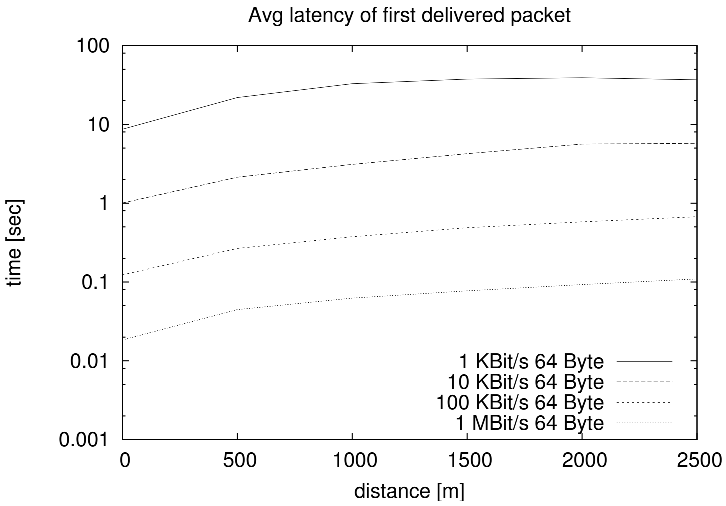


Figure 3.8: Average latency of delivered warning messages versus distance of communication partners.

driver it seems likely that at least a bandwidth of 100 KBit is needed. Here the duration to transmit a packet grows from 0.1 s in a distance to 500 m up to almost 1 s in a communication range of 2 500 m.

The results are, however, strongly affected by the considered scenario and dissemination application and are thus only partially suited for technical conclusions regarding this VANET application. Though, it is certifiable that by using the simulator coupling new and different insights in these applications can be achieved. We will thus use this simulator environment for the evaluation of the protocols presented in this thesis.

Chapter 4

Feasibility of Information Dissemination

In the context of vehicular ad-hoc networks (VANETs), a number of safety and convenience applications have been proposed. Many of them rely on distributing data, e. g., on the current traffic situation, or on free parking lots. Often, the data needs to be distributed over long distances, for example to allow a driver to choose between different arterial roads when driving into the city center. Typically, the applications are based on some form of proactive information dissemination. Although a variety of optimizations is possible, the basic idea of such a dissemination scheme is that every node maintains a local knowledge base, where it stores information, e. g., on road conditions or parking lot occupancies. The nodes periodically single-hop broadcast all or parts of their knowledge base to their neighbors. Upon reception, a node integrates new or updated information into its local knowledge base. Step by step a local overview of the total scenario emerges.

In this chapter, we tackle a fundamental question, which is highly relevant for all such applications: is the required data dissemination feasible at all, and what requirements need to be fulfilled to make it work? While it is fairly obvious that information dissemination will work well when all or nearly all vehicles participate in the vehicular ad-hoc network, this is not at all self-evident during the early rollout of VANETs, where the number of equipped vehicles is small [MMP$^+$05].

The focus of this study is on city scenarios, where the environment is rather complex and many of the proposed application types are particularly useful. While, e. g., up-to-date traffic information on the comparatively small number of highways could also be collected at a central point and distributed via wireless infrastructure, like UMTS or satellites, the detailed, geographically small-scale, and continuously updated information that is necessary for city environments may stress centralized approaches beyond their limits.

We use the specialized simulation environment mentioned above for an inner-city VANET scenario, in order to evaluate the performance that a dissemination protocol can achieve. In particular, we look on upper bounds for metrics like the speed and efficiency of the information dissemination, depending on the number of equipped vehicles on the road. By means of an idealized stub dissemination protocol, we are able to evaluate the general feasibility, independent from a specific protocol or application. Because the simulation scenario we use is closely modeled after a real city, it allows us to give concrete numbers on the necessary amount of vehicles provided with vehicle-to-vehicle communication equipment in order to achieve a certain performance. Our results demonstrate that, in particular during the initial rollout phase of VANET technology, information dissemination is not practically feasible without the use of supporting infrastructure. On the other hand, however, we also show that a limited amount of simple and relatively inexpensive infrastructure based devices can be the critical factor that improves the situation significantly, and allows to build a working dissemination-based application.

The remainder of this chapter is structured as follows. In Section 4.1, some theoretical considerations are presented, leading to general insights on how information dissemination happens in an inner-city VANET, and where the limiting factors are. The simulation environment used for the evaluation is described in Section 4.2, preliminary results of our simulation study are presented in Section 4.3. In Section 4.4, we introduce the concept of additional infrastructure based devices. We show in Section 4.5 how the preliminary results can be significantly improved by this infrastructure support even if only few cars are equipped. Finally, we conclude this chapter in Section 4.6.

4.1 Connectivity in VANETs

Let us consider an application for VANETs that uses proactive data dissemination in a city environment. We now look at the connectivity that can be expected if the density of equipped vehicles is low. The network connectivity is a limiting factor for information dissemination. A low connectivity of the network may have serious effects on the dissemination speed, and thus on the up-to-dateness of the information. It also determines how long it takes for a vehicle entering the VANET until it meets other participating vehicles and receives any information at all[1].

Data can be passed on from vehicle to vehicle by wireless communication, or it can be carried around by a car, and is thus transported with the car's locomotion. Both ways allow the information to reach different network areas, and in practice both will coexist. However, the possible dissemination speed is much higher for wireless communication.

[1] In another perspective this is related to the latency of information dissemination which is studied in [KY08].

Data transport via locomotion happens at the car's speed, in cities typically at most 50 km/h. The propagation speed via wireless communication along a chain of equipped vehicles, where each one is within the communication range of its predecessor, can be approximated as follows. Let us consider a number of equipped vehicles driving at distances in the order of a typical radio communication range between two consecutive cars, like 250 m. Before newly arrived information is propagated further by a car, there will be some delay. This delay mainly occurs because it takes some time until the next broadcasting interval has elapsed, but other factors like, e. g., backoff times also add up to this. The transmission time itself is negligibly small in comparison. Even if we estimate the delay pessimistically to be one second, we get a data propagation speed along the chain of one radio range per second, which is 900 km/h.

Furthermore, in contrast to data dissemination by locomotion, wireless propagation is possible in and against the driving direction. This becomes relevant when taking into account that cars are particularly interested in information on the areas ahead of them. Considering cities, there are often asymmetric traffic situations, where many vehicles are driving in the same direction, e. g., towards the commercial quarters in the morning rush hour, while the oncoming traffic is relatively sparse. So, few cars drive in the more important dissemination direction, and thus data transport by locomotion will work particularly bad.

These simple reflections demonstrate that the formation of chains of equipped vehicles, each driving within the radio range of its respective predecessor, is essential for a satisfying dissemination performance. Thus, we are now interested in the probability that such a chain of equipped vehicles exists on a road. Assume that the distances between equipped cars are exponentially distributed and pairwise independent. This is a standard assumption [RML02]. It is optimistic in the case of city scenarios, as will soon become clear, but it serves well for a best-case estimation.

Let r be the radio communication range, and let ρ denote the average number of equipped vehicles per radio range of road. We call ρ the *equipment density*. Let d_i be exponentially distributed, pairwise independent random variables. d_i stands for the distance between the i-th equipped vehicle and its successor. The parameter λ of the exponential distribution of the d_i is chosen as $\lambda = \rho$, so the expected equipment density of our chain matches ρ. The probability that n consecutive equipped vehicles drive at distances of less than r, can now be calculated as

$$P_v(\rho, r, n) = \prod_{i=1}^{n-1} P(d_i \le r) = \left(1 - e^{-\rho r}\right)^{n-1}. \tag{4.1}$$

For disturbed traffic, which is common in cities, cars tend to form clusters on the road. Consider for example a traffic light, where a number of cars queue and then continue driving closer together, in a cluster. In this case the assumption of exponentially distributed inter-arrival times does not apply. Then, the above estimation tends to be too optimistic: at the same average equipment density, the formation of clusters means that longer gaps—between the clusters—become more probable. Therefore, a non-connected situation can be expected to be even more likely.

This allows us to calculate the expected length of such a chain of equipped, connected vehicles:

$$\begin{aligned}
E[l] &= \sum_{n=2}^{\infty}\left(\prod_{i=1}^{n-1} P\left(d_i \le r\right)\right) P\left(d_n > r\right) \sum_{i=1}^{n-1} E\left[d_i \mid d_i \le r\right] \\
&= \sum_{n=2}^{\infty}\left(1-e^{-\rho r}\right)^{n-1} e^{-\rho r}(n-1) \int_0^r x \frac{\rho e^{-\rho x}}{1-e^{-\rho r}}\, \mathrm{d}x \\
&= \left(\sum_{n=1}^{\infty} n\left(1-e^{-\rho r}\right)^n\right) e^{-\rho r} \frac{1}{1-e^{-\rho r}}\left(\frac{1}{\rho}-\left(r+\frac{1}{\rho}\right) e^{-\rho r}\right) \\
&= \frac{1-e^{-\rho r}}{e^{-2\rho r}} e^{-\rho r} \frac{1}{1-e^{-\rho r}}\left(\frac{1}{\rho}-\left(r+\frac{1}{\rho}\right) e^{-\rho r}\right) \\
&= \frac{1}{\rho e^{-\rho r}}-r-\frac{1}{\rho}
\end{aligned} \tag{4.2}$$

Assuming a sufficiently long road segment of length L and a sufficiently large equipment density ρ, the number of equipped cars on the road segment can be approximated as $\rho \cdot L$. Again for exponentially distributed inter-vehicle distances, the probability P_c that radio connectivity exists and thus fast dissemination by multihop wireless communication over the whole distance L can happen at some time instant is then

$$P_c(\rho, r, L) \approx \prod_{i=1}^{\lfloor \rho L \rfloor} P(d_i \le r) \approx \left(1-e^{-\rho r}\right)^{\rho L}. \tag{4.3}$$

This means that the probability of continuous connectivity decreases exponentially over an increasing distance L. Therefore, multihop radio transport alone will *not* be sufficient, and data transport via locomotion is an important additional factor, as long as the equipment density does not become very high. The interplay of the two ways of data transport determines the achievable performance of data dissemination in a city environment.

The effects of these characteristics are depicted in Figures 4.1 and 4.2. It is quite obvious that information dissemination solely by chains of equipped vehicles is far from sufficient at equipment densities as they occur during VANET rollout.

4.2 Simulative Evaluation Methodology

In the following section, we present a simulation study on the feasibility of information dissemination in a city environment. It is carried out using a VANET simulation environment. We now introduce this environment as well as a stub application that we have used.

4.2.1 Simulation Environment

For the simulation based evaluation we make usage of the simulator coupling presented in Chapter 3. Vehicular movements are generated by the microscopic traffic simulator VISSIM [PTV].

We use a traffic model of the extended downtown area of Brunswick, Germany. This traffic model covers a geographical area of about $16 \times 16\,\text{km}^2$, with more than 500 km of roads and up to 10 000 vehicles. The vehicular traffic in the model is based on extensive measurements taken by the city administration of Brunswick for the purpose of traffic planning. It models the time between 6:00 am and 10:00 am. VISSIM is coupled with the well-known network simulator ns-2 [ns2] in version 2.29. The combination of VISSIM and ns-2 allows to conduct a detailed simulation of both, vehicle movements and network events.

In ns-2, we use the two-ray ground propagation model with a communication range of 250 meters and a carrier sense range of 550 meters. The network simulator is enhanced with the obstacle modeling presented in Section 3.2 that does not allow radio signals to propagate through the walls of buildings. IEEE 802.11 is employed as the MAC protocol.

4.2.2 Equipment Density

Since we are interested in the influence of the amount of car-to-car enabled vehicles on the protocol performance, the question arises which metric one should use to describe this factor. The *penetration ratio* (or *market penetration*) is commonly used. It is defined as the percentage of equipped vehicles, out of all vehicles.

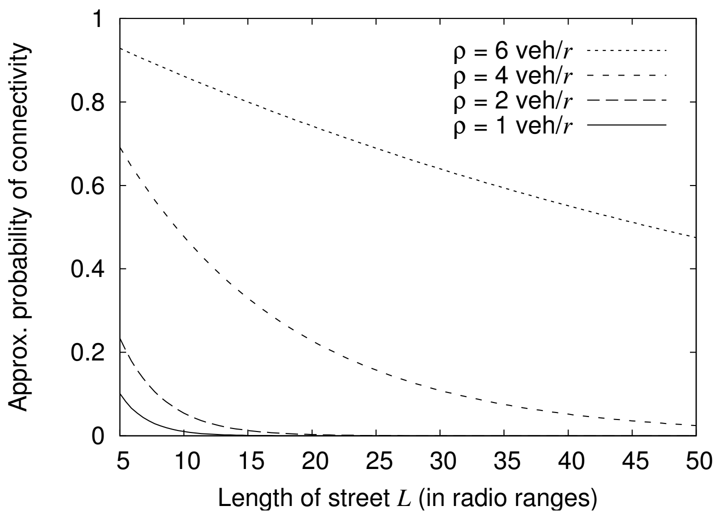


Figure 4.1: Approximated probability of radio connectivity dependent to the street's length.

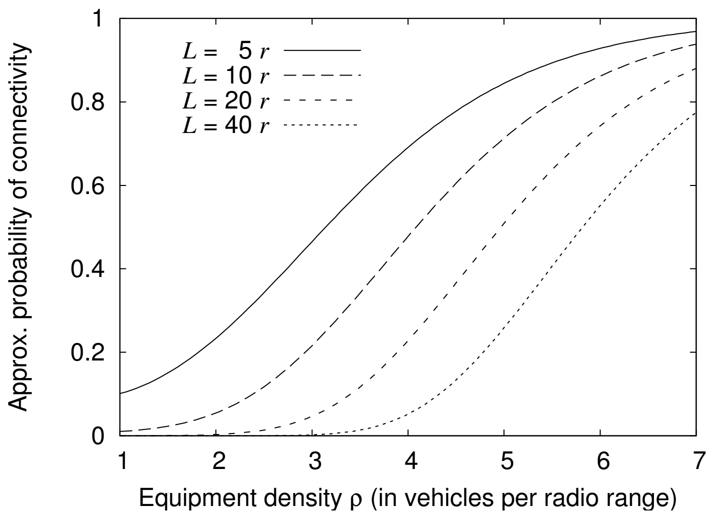


Figure 4.2: Approximated probability of radio connectivity dependent to the equipment density.

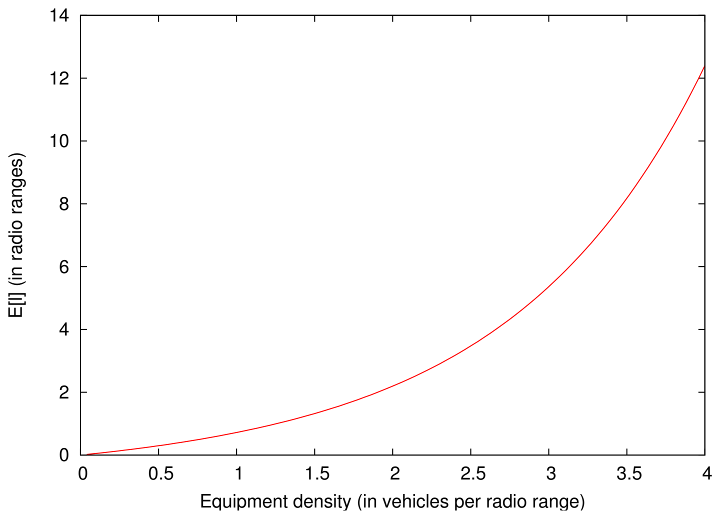


Figure 4.3: Estimated length of chain dependent to the equipment density.

We, however, consider the penetration ratio inappropriate for our purposes. The number of cars on the road changes largely over time. In the night it can be orders of magnitude lower than during rush hour. This means that a protocol can work very well at a low penetration ratio during rush hour, since still many equipped cars are on the road. On the other hand, even a very high penetration ratio can be insufficient if the traffic volume is low. Therefore, we use the *equipment density* ρ. It is, as defined above in Section 4.1, the number of vehicles participating in the VANET per radio range of road. The equipment density is independent from the total traffic volume.

In the Brunswick model, a penetration ratio of 100 % with a radio communication range of 250 meters corresponds to an average equipment density between 2.25 and 5 vehicles per radio range, depending on the simulated time of day. However, the inhomogeneous distribution of vehicles has to be considered, so the local equipment density at some point can be higher or lower.

4.2.3 Considered Application

We consider a simple stub dissemination application for our simulations. Whether information on one single data source or on many of them is disseminated influences the amount of information that is redistributed, and thus the necessary network bandwidth. It does not, however, affect the network connectivity. Regarding the utilized network bandwidth, many optimizations, e. g., by using data aggregation strategies, are possible. Optimizing the network connectivity by means of protocol design is not as easily possible. This is why we concentrate on the latter aspect here. Consequently, for our purposes, it is sufficient to use only one single data source.

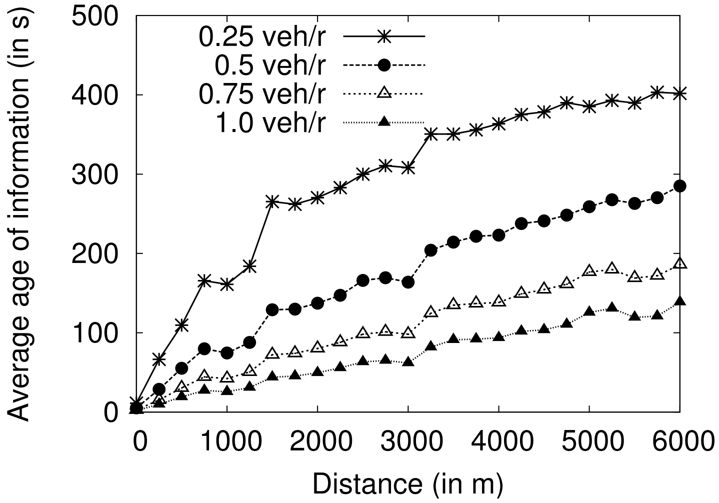
We place this data source in the city center of the scenario. Each car passing by will make an observation by measuring the current "value" of this data source, which is simply a timestamp. This information is then proactively disseminated by periodic broadcasting, as described above. The vehicles use a periodic broadcasting interval of one second, the broadcasted packets have a size of 1 KB. Whether information is available in which vehicle at which point in time, and how old this information is, allows us to draw conclusions on how well information can be disseminated in a vehicular ad-hoc network.

4.3 Preliminary Results

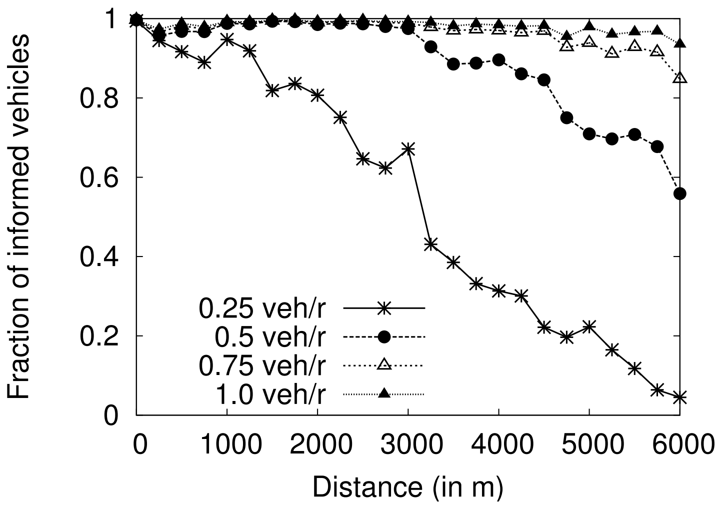
For the evaluation of information dissemination in VANETs, we have performed simulations over a wide parameter range. Figure 4.4(a) shows the average age of information available in a vehicle as a function of the distance from the data source. It can be seen that the age of the information grows approximately linearly with the distance. The propagation speed rises significantly with an increasing equipment density. This is because the probability of the formation of long chains increases with a growing equipment density. This in turn means that data transport via wireless communication becomes more and more predominant, while the importance of transport via locomotion decreases.

At a low equipment density, the average duration for the dissemination of information to the outer areas can reach a value as high as 400 s. But a low equipment density has an even more serious influence on the probability of a vehicle knowing anything about the data source *at all*. This statistic, after 500 s of simulation time, is depicted in Figure 4.4(b). The probability to obtain information at an increasing distance decreases rapidly. The main reason for this trait is that it takes some time after a vehicle starts its trip from a residential area, a parking lot, etc., until it meets another car from which it can obtain information. From the results it can be seen that a relatively high equipment density is necessary for sufficiently reliable information dissemination. Consequentially, the rollout phase of a VANET application, where only a low equipment density is available, is highly problematic.

In order to gain more information on the exchange of information within a VANET, we evaluated in which geographical regions the highest number of successful information transfers happen. A successful information transfer takes place whenever a car receives a beacon that contains more up-to-date information than it already had in its knowledge base. Figure 4.5 shows the city of Brunswick on the x- and z-axis. The small-scale density of successful information transfers is plotted on the y-axis. Because only the relative



(a) Average age of information.



(b) Fraction of informed cars.

Figure 4.4: Dissemination without infrastructure support.

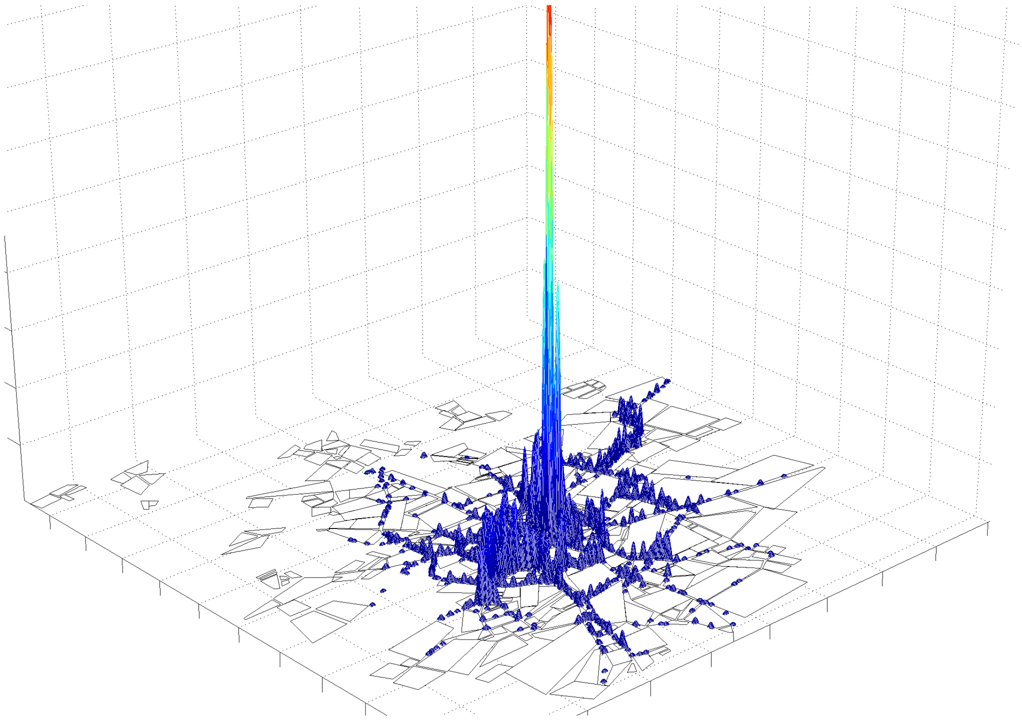


Figure 4.5: Geographical distribution of successful information transfers.

heights of the peaks are relevant here, we have left out the axis labels, keeping the figure concise. It can be seen that, apart from the inner city where the information originally stems from, most successful information transfers occur in few, geographically limited areas. These are located mainly at the ends of and along the main arterial roads. It is here where cars with different knowledge meet. We therefore call these areas *communication market places*. They are vital for successful information dissemination.

4.4 Stationary Supporting Units

4.4.1 Idea

The results presented above clearly show that during the rollout of VANET technology some kind of support is needed. Otherwise, many envisioned applications are unlikely to work until a large fraction of vehicles participates. We therefore propose to install relatively inexpensive, stationary devices at strategic positions. These *Stationary Supporting Units (SSUs)*—sometimes also called Roadside Units (RSU), Information Kiosk, or simply

Supporting Units (SU)—participate in the VANET. They collect and redistribute information, thereby leveraging the dissemination in the network.

The SSUs may either be stand-alone units, or they can be connected over some backbone network. The computational and memory resources needed for a SSU and therefore their hardware costs are very limited. While stand-alone SSUs are particularly cheap, networking them implies additional expenses for the backbone connection, which might be realized, e. g., via wireless infrastructure networks like WiMAX, GSM, GPRS, or UMTS. Because of the higher costs per unit the possible number of networked SSUs is much more limited than that of the non-networked variant.

Just like the cars themselves, supporting units receive information from the VANET and periodically rebroadcast their knowledge. The most important difference to vehicles is that SSUs do not move. Besides that, networked SSUs share a common knowledge base. This means that information learned by one SSU may be rebroadcasted by all of them.

4.4.2 Positioning of Supporting Units

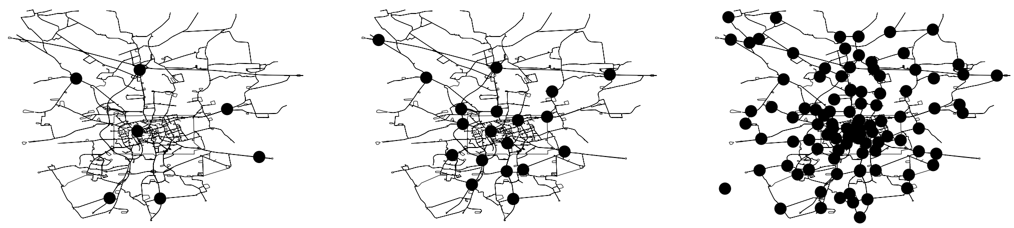
A central question that now arises is where to position the SSUs, in order to allow for a best-possible support of the VANET. In the following paragraphs, we analyze different heuristics of the positioning of SSUs, and we assess whether networking the SSUs is worth the additional effort, i. e., whether few networked or many non-networked SSUs perform better. We concentrate on three possible strategies for positioning the stationary supporting units:

At Market Places The identification of communication *market places* led us to the idea to install the SSUs there. Since many cars learn new information at the market places, in particular networked SSUs promise to achieve that the information is as up-to-date as possible. There is a small set of clearly predominating communication market places. So, in our first strategy, the very limited number of seven SSUs is installed. One is located in the city center, the others at the most predominant information market places in the periphery. We assess SSUs at communication market places both stand-alone and networked.

At High Traffic Density Areas The communication market place strategy is based on the observed communication pattern in the network. It is also possible to select strategically promising positions based directly on the vehicular traffic pattern. In our second strategy, SSUs are installed at *high traffic density areas*, along the main roads. There, typically relatively few successful information transfers happen, but SSUs might assist to bridge gaps between chains of vehicles by storing the information and passing it on when the next vehicles arrive. Since the high traffic density areas outnumber the communication market places, this strategy uses more supporting units. We use 19 SSUs in our simulations, and again simulate them stand-alone and networked.

Randomly Distributed As a last heuristic we have placed 100 SSUs *randomly* within the road network. A that high number of SSUs can potentially improve the connectivity of the network significantly. However, the effort of installing that many SSUs is only feasible without a backbone connection for each one. So, this strategy is simulated only for non-networked SSUs.

The positions of the supporting units in Brunswick's road network for all three positioning strategies are depicted in Figure 4.6.



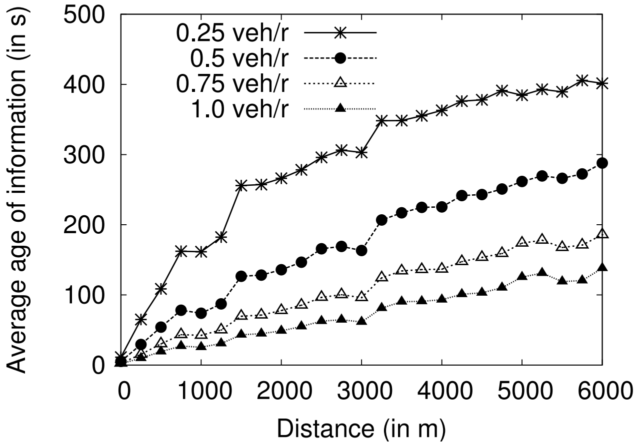
(a) At market places. (b) At high traffic density areas. (c) Randomly distributed.

Figure 4.6: Positioning of stationary supporting units.

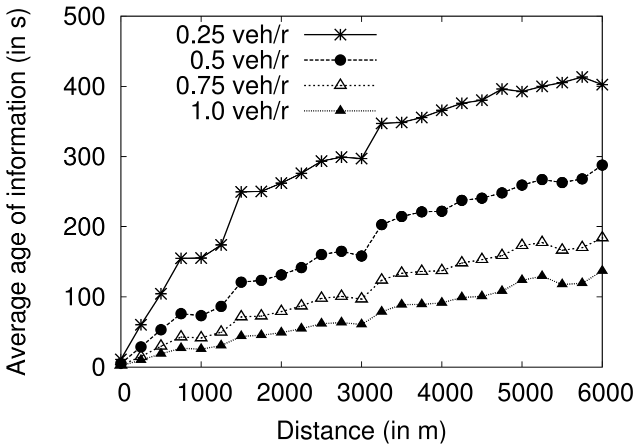
4.5 Simulation Results

Figures 4.7 and 4.8 show the results of the simulations with stand-alone SSUs. In comparison to the results without any supporting units, the information age and the number of informed cars improve only slightly, even for the very high number of SSUs in the random positioning heuristic.

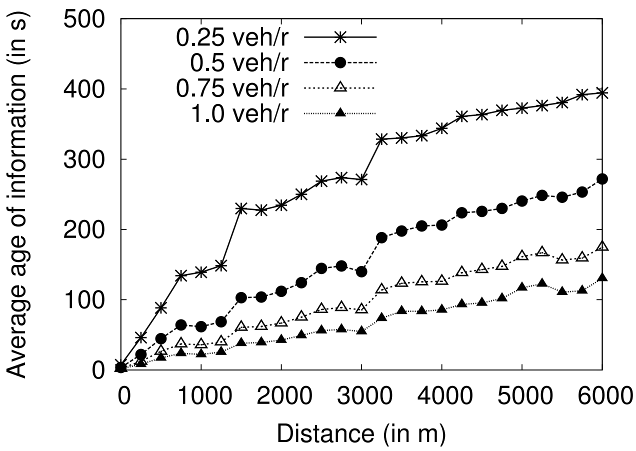
In contrast, Figures 4.9 and 4.10 show the advantages of networked supporting units. Compared to the results without or with stand-alone SSUs, a significant improvement can be achieved, especially in regions far away from the information source. The improvement is also large for low equipment densities and few SSUs. In particular, as depicted in Figures 4.9(b) and 4.10(b), the fraction of informed cars increases dramatically. During the rollout phase this can make the difference between a well-working and a non-working service.



(a) At market places.

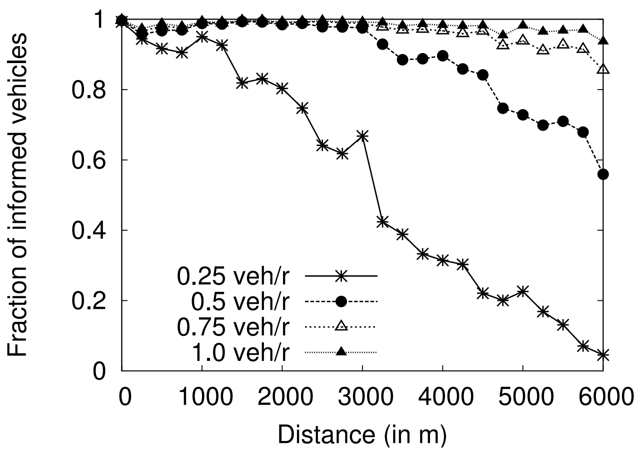


(b) At high traffic density areas.

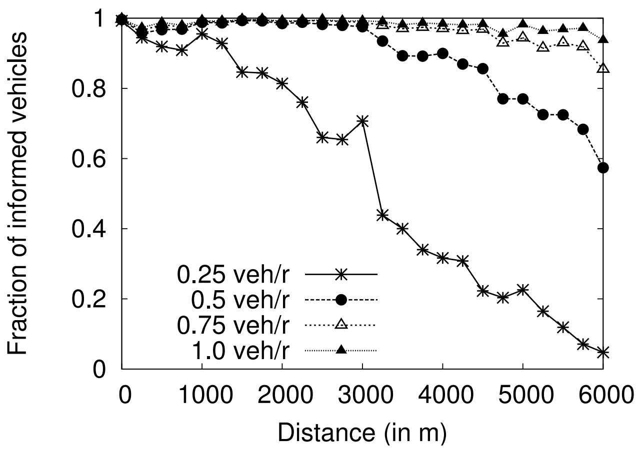


(c) At randomly chosen locations.

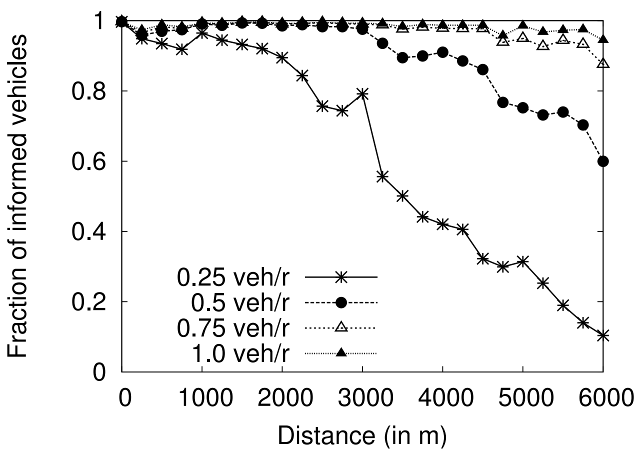
Figure 4.7: Stand-alone stationary supporting units—Average age of information.



(a) At market places.

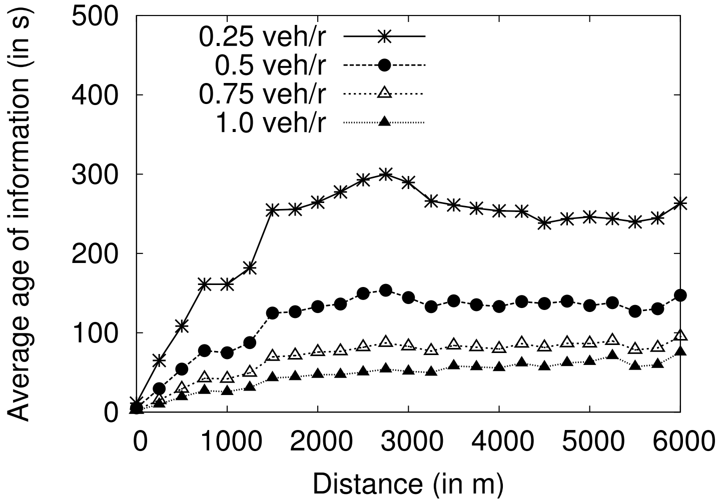


(b) At high traffic density areas.

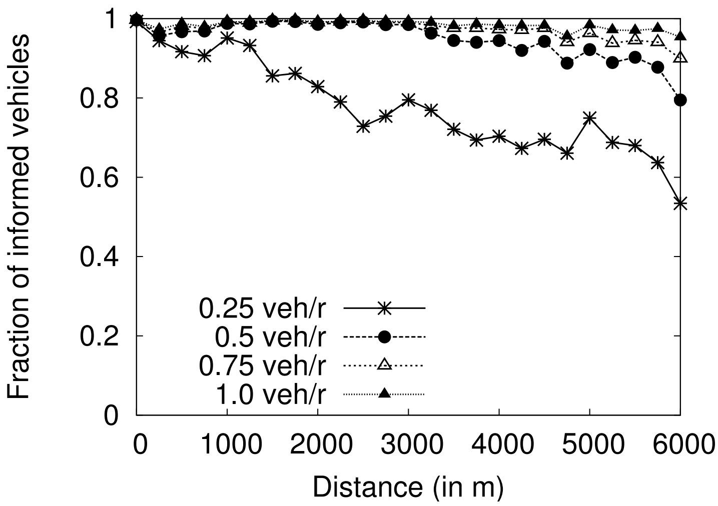


(c) At randomly chosen locations.

Figure 4.8: Stand-alone stationary supporting units—Fraction of informed cars.

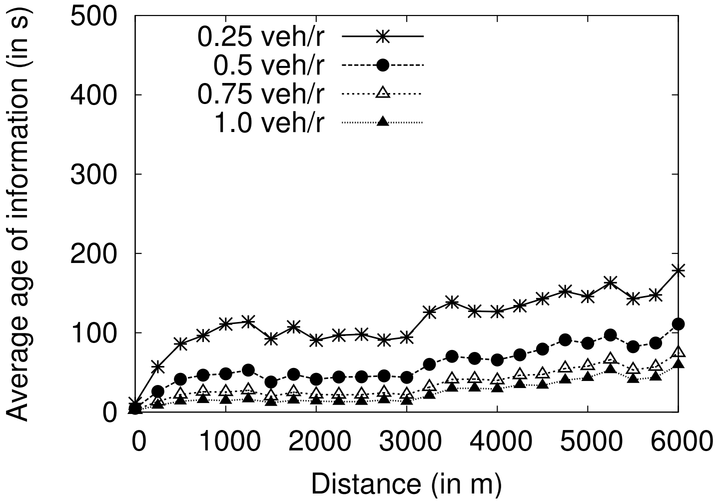


(a) Average age of information.

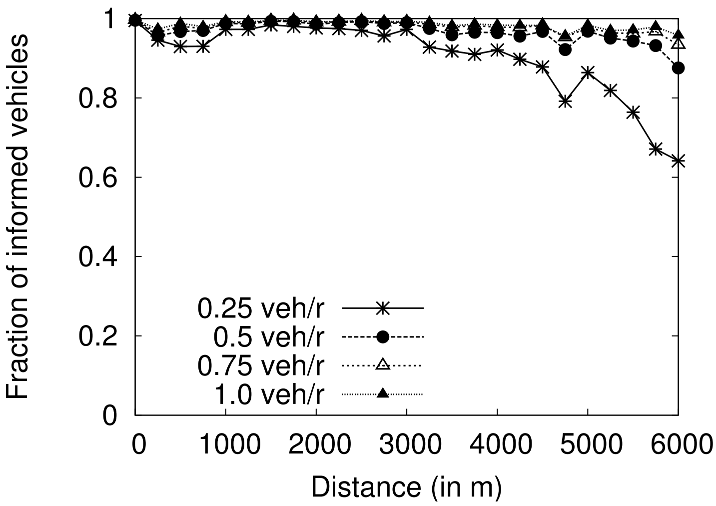


(b) Fraction of informed cars.

Figure 4.9: Networked stationary supporting units at information market places.



(a) Average age of information.



(b) Fraction of informed cars.

Figure 4.10: Networked stationary supporting units at high traffic density areas.

All of the figures presented so far describe the results in dependency of the distance from the data source. But they do not show, however, the geographical distribution of the information age.

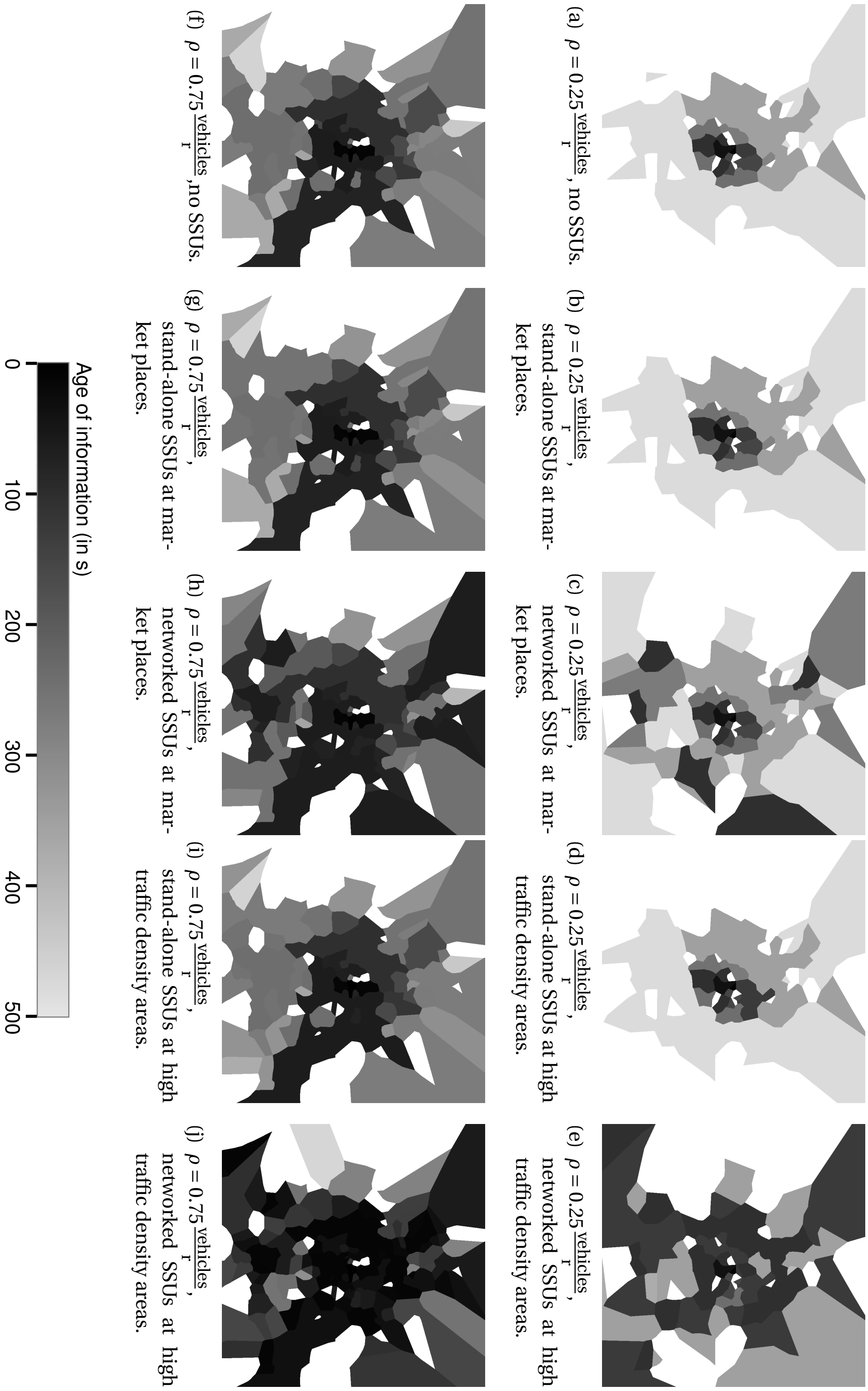
In Figure 4.11, the information age after 500 s of simulation time at each point of the scenario is depicted. The figure is a Voronoi diagram, i. e., each point is colored according to the closest vehicle. The same shade indicates the same age of information. The darker an area is, the more up-to-date is the available information. White areas indicate that no information is available.

The positions of the stationary supporting units, which are located according to Figure 4.6, can easily be spotted, for instance, in Figure 4.11(c). Vehicles with most current information—indicated by the dark areas—can be found in the close vicinity of such a supporting unit.

In these plots, the strong dependency of the age of information on the equipment density is once again visible. Equipment densities of approximately 0.25 vehicles per radio range without stationary supporting units show an acceptable coverage only of the downtown area. In regions further away, the information becomes more and more outdated, and more and more vehicles are completely uninformed. Stand-alone supporting units cannot improve this situation significantly, so the insights of the above presented results are confirmed.

It is evident that networked SSUs influence the up-to-dateness of information beneficially. This is even true if the equipment density is low, and if only very few supporting units are used. Nevertheless, most vehicles can be informed quickly and accurately with both analyzed positioning strategies. But in particular with the slightly higher number of SSUs at high traffic density areas, an increase in the performance of the dissemination process can be reached.

Finally, in the Figures 4.11(f)–4.11(j), it is depicted the effects of information dissemination with higher equipment densities. Here, the density corresponds to approximately 0.25 vehicles per radio range. However, the figures underline that at higher equipment densities the information dissemination starts working much better. Nevertheless, the dissemination process is still largely improved by networked stationary supporting units.



(a) $\rho = 0.25\,\frac{\text{vehicles}}{\text{r}}$, no SSUs.

(b) $\rho = 0.25\,\frac{\text{vehicles}}{\text{r}}$, stand-alone SSUs at market places.

(c) $\rho = 0.25\,\frac{\text{vehicles}}{\text{r}}$, networked SSUs at market places.

(d) $\rho = 0.25\,\frac{\text{vehicles}}{\text{r}}$, stand-alone SSUs at high traffic density areas.

(e) $\rho = 0.25\,\frac{\text{vehicles}}{\text{r}}$, networked SSUs at high traffic density areas.

(f) $\rho = 0.75\,\frac{\text{vehicles}}{\text{r}}$, no SSUs.

(g) $\rho = 0.75\,\frac{\text{vehicles}}{\text{r}}$, stand-alone SSUs at market places.

(h) $\rho = 0.75\,\frac{\text{vehicles}}{\text{r}}$, networked SSUs at market places.

(i) $\rho = 0.75\,\frac{\text{vehicles}}{\text{r}}$, stand-alone SSUs at high traffic density areas.

(j) $\rho = 0.75\,\frac{\text{vehicles}}{\text{r}}$, networked SSUs at high traffic density areas.

Figure 4.11: Geographical information dissemination after 500 seconds.

4.6 Conclusion

The theoretical results as well as the simulation results presented in this study deal with the information dissemination of VANETs in a city environment, and show general limitations of the approach. The simulation results of a city scenario indicate that a dissemination application can operate satisfactorily starting from equipment density averages of around 0.25 vehicles per radio range. In the examined scenario this corresponds to a penetration ratio of approximately 5 % during rush hour (at 8:00 am), with a radio range of 250 m.

We identified the formation of chains of equipped vehicles as a vital mechanism to ensure fast information dissemination. Along these chains, information can be transmitted much faster than by carrying the information in a vehicle. With rising equipment density the probability for the formation of long chains, and thus their effectiveness, increases.

During VANET rollout, stationary supporting units can be used to facilitate the information dissemination, in particular for data transport against the main direction of traffic. To achieve this goal, the SSUs need to be networked and need to share a common knowledge base. Then, they achieve both a higher probability to receive information at all, and more up-to-date information.

Chapter 5

Capacity Restrictions of VANETs

Research on Vehicular Ad-hoc Networks (VANETs) aims to improve road safety, traffic efficiency, and driver convenience. Typical examples in the area of efficiency and convenience applications are cooperative traffic information or parking guidance systems. This class of applications typically requires to distribute dynamic information (traffic condition, free parking slots,...) from many points (road segments, parking places,...) to many or all nodes in a large network area.

Because wireless multihop networks in general [GK00] and VANETs in particular have very limited capacity, it is obviously not possible to send continuous updates about each geographical point to all network participants. It has therefore been proposed to aggregate information more and more with increasing distance, i. e., to distribute coarse information about large, distant regions, and more detailed information about smaller areas in the closer vicinity. Basically, aggregation is a method to cope with the problem of limited network capacity. Here, we estimate the amount of information transmitted by a typical VANET application and propose a scheme fitting these information in the given "physical" capacity. A change of the lower layer technology will thus not influence the application itself.

By taking these properties into account, we analyze the theoretical limits of a distributed information dissemination scheme for a VANET. We do so in a very abstract model of cooperative VANETs, and look at the asymptotic scaling behavior. Our primary focus is on the minimum aggregation requirements for scalable dissemination applications.

The remainder of this chapter is structured as follows. In Section 5.1 we introduce the considered model of data dissemination. We deal with the sources of data and where this data is requested by some nodes. In Section 5.2 we determine the lower bound of an aggregation scheme for cooperative traffic information systems in a wireless, multihop environment. Finally, in Section 5.3, we conclude this chapter by summarizing the key results.

5.1 Model

Our model represents the "world", i. e., the area the system is deployed on, by the real plane $\mathbb{R}^2$. For practical purposes, this is a reasonable approximation of a city area, country, or even continent. On this plane, there is a set $M \subset \mathbb{R}^2$ of points at which information can be obtained through measurements. These *measurement points* could, for example, represent all the street segments, for which passing-by cars could observe the current traffic density, driving velocity, number of free parking places, road surface condition, and so on. The observed values are time-varying, i. e., the measurements are always made at some specific time instant. Due to this temporal property, a measurement point can be seen as an information source, which "produces" information about the measured value whenever cars make measurements. The task of an information dissemination protocol is to deliver this generated information to the interested network participants. The focus here is to assess the asymptotic limits of such protocols with respect to the *bandwidth* at which information about each single measurement point, a car is interested in, can be delivered to this car.

Generally, the network capacity limits will depend on the distribution and density of the measurement points on the plane. If there are only few measurement points, disseminating the information will be easier than in the case of a large number of information sources. If we allow for arbitrarily large and dense clusters of measurement points, an arbitrarily large amount of information can be generated in a very limited area; then, their information can obviously not be communicated even locally and capacity considerations become meaningless. However, since we intend to stay as generic as possible in our analysis, we impose only a very weak condition on the distribution of the measurement points. We concentrate on the case where the measurement point distribution satisfies a *max-density condition.* This condition essentially states that there are no arbitrarily large, arbitrarily dense groups or clusters of measurement points. It is formally defined as follows:

Definition 1. *A set of measurement points M fulfills a max-density condition with parameters $\delta > 0$ and $r_0 > 0$ if and only if for any circle on $\mathbb{R}^2$ with radius $r \geq r_0$ the number ν of measurement points that lie within the circle is bounded above by*

$$\nu < \delta r^2. \tag{5.1}$$

Note that in the previous definition, parameter choices where $\delta r_0^2 \leq 1$ do not make sense, because they would not allow for even just one single measurement point to exist: let m be a measurement point and consider a circle with radius $r = r_0$ around m; this circle would contain $1 \geq \delta r^2$ measurement point and would thus already violate the max-density condition. Therefore, we may safely assume that δr_0^2 will always be larger than 1.

Intentionally, our model does not constrain the possible distribution of interests in the system in any way, because we are interested in fundamental limits that hold for any application, whereas the specific structure of the interests is highly application dependent. We model the interests of the participants in the dissemination system by a set $\mathcal{I}$ of pairs (x, m), where in each pair x is the position of a network participant that is interested in data from a measurement point $m \in M$. Consequently, the formal definition is as follows.

Definition 2. *Let the interest set $\mathcal{I}$ be an arbitrary subset of $\mathbb{R}^2 \times M$.*

In a VANET, vehicles could for instance be interested in information describing the traffic situations along possible routes, to determine the currently fastest one.

Definition 3. *The* distance *of an interest $i = (x, m) \in \mathcal{I}$, denoted by $\|i\|$, is the distance between the position of the interested party x and the measurement point m, i. e.,*

$$\|i\| := \|x - m\|. \tag{5.2}$$

In a network with an unlimited capacity, it would not be a problem to deliver all measurement data about all measurement points to each interested participant. In practice, however, each network imposes limitations on the maximum bandwidth between communication partners. Again, we aim to capture the essence of these limitations in a way that is as generic as possible, with as few specifics of and assumptions about any particular network or communication mechanism. Our formulation for the transport capacity limits is based on the fact that between disjoint partitions (or, areas) of the network, the available bandwidth may not be arbitrarily high. We observe that if we, again, place an arbitrary circle on the plane, then the whole information transport between the areas within and outside of the circle must necessarily cross the circle boundary.

Now we assume that the "density of communication links" crossing the circle boundary cannot become arbitrarily high. For instance, in the case of wireless communication and VANETs, the number of parallel transmissions into (or out of) the circle is clearly limited by interference and maximal density of cars on the circle boundary. Consequently, there is an upper bound on the communication bandwidth over the circle border that will grow linearly with the circumference of the circle. We capture this formally in the following way.

Assumption 1. *We assume that there are a minimum radius* $r_1 > 0$ *and a constant* $\xi > 0$ *such that, given an arbitrary circle with radius* $r \geq r_1$*, the maximum possible communication bandwidth b between the inside and the outside of the circle is bounded above by*

$$b < \xi r. \tag{5.3}$$

5.2 Bandwidth Characteristics of Scalable Dissemination Schemes

Before we turn towards the question of *how much* aggregation is necessary, let us first argue why data aggregation is necessarily needed for dissemination services in VANET *at all*. The bandwidth at which each network participant may received data from measurement points is limited. According to our model, considering a circle with radius r_1 around the participant's current position, it is clear that the total data rate for information from measurement points outside this circle provided to the participant cannot exceed ξr_1. If each measurement point produces data with at least some positive minimum data rate b and the participant is interested in more and more measurement points outside the circle, it is immediately evident that the necessary total bandwidth will at some point exceed ξr_1. Thus, in order to be able to serve the interests of the network participants under all circumstances, a VANET dissemination protocol *must* use techniques to reduce the necessary bandwidth.

One approach would of course be to limit the number of interests. This, however, is hardly viable. It is of course easily possible to put a limit on the allowed number of interests of a single network participant; the number of interests may even be inherently constrained by the application. Note, however, that it will often be the case that many network participants which are interested in very different parameters are located in the same geographical area. For example, cars with very different destinations and driving routes may be underway on the same road. They will be interested in, for instance, the current traffic condition on very different parts of the road network. However, despite the almost arbitrarily large variety of interests, the total ingress bandwidth for these cars is still limited. (In terms of our model, consider a circle encompassing all the cars, then the total bandwidth into this circle is limited.) Thus, a much more promising approach is to use in-network summarization and aggregation techniques to adjust the resolution of the provided information (and thus the necessary bandwidth) in order to respect the inherent bandwidth limitations of inter-vehicle communication.

Such aggregation techniques are possible in many different shades and flavors. As mentioned in Sections 2.3.3 and 2.3.4, for instance, information from the measurement points within the same geographical area may be aggregated, such that only a summary is transmitted to interested parties further away, proposed in [CGM06]; information from individual vehicles on a road may be combined as in [NDLI04b]; the frequency at which information about certain geographical regions is transmitted may be adjusted like in [KSA02]; syntactic compression techniques may be used to yield smaller aggregates, as suggested in [IW08a]. All these techniques—for themselves or in combination—aim at reducing the bandwidth used per measurement point with increasing distance of the interest, either by using coarser approximations of the value itself (i. e., data representations requiring less bits), by sending updates less often, or by transmitting one single value for a whole group of measurement points.

In our analytical approach to determine the fundamental limits of aggregation we need to find a way to abstract from the concrete approaches and mechanisms of aggregation. We therefore identify the reduction of the bandwidth per measurement point with increasing distance as a common feature, which is present regardless of the concrete means to achieve it. Detailed, frequently updated information is made available for nearby areas, while the representations become coarser for longer distances. From this point of view, the essential feature of an aggregation scheme is the amount of bandwidth spent depending on the distance: given an interest with distance d, how much bandwidth is spent on making information available to the interested party? Consequently, we characterize an aggregation scheme by its *bandwidth profile*.

Definition 4. *A monotonically decreasing function*

$$b\colon \mathbb{R} \to \mathbb{R}_{\geq 0} \tag{5.4}$$

is a bandwidth profile of an aggregation scheme $\mathcal{A}$ if $\mathcal{A}$ ensures that for all interests $i = (x, m) \in \mathcal{I}$ the interested party x is supplied with information about measurement point m at least with data rate $b(\|i\|)$.

In the remainder of this section, we will prove for the general case of arbitrary measurement point sets (with a max-density property) and interests that if an aggregation scheme has a bandwidth profile b for which $b(d) \notin o(1/d^2)$[1], then the bandwidth limitations established by Assumption 1 do not allow to ensure that all interests can be appropriately served. Hence, practical aggregation schemes should be constructed in a way such that the bandwidth used per measurement point decreases faster than with the square of the interest's distance.

[1] $o(1/d^2)$ describes the set of functions which decrease asymptotically faster than $1/d^2$. More formally, $b(d) \in o(1/d^2)$ if and only if

$$\forall\, c > 0 : \exists\, d_0 > 0 : \forall\, d > d_0 : b(d) < \frac{c}{d^2}.$$

The following lemma states a sufficient (but not necessary!) condition for a measurement point distribution to ensure that a max-density condition holds.

Lemma 1. *Let $\delta > 0$, $r_0 > 0$ be given such that $\delta r_0^2 > 1$. If the measurement points in M are distributed in such a way that the distance between any two measurement points is at least*

$$\Delta := \frac{2}{\sqrt{\delta} - r_0^{-1}}, \tag{5.5}$$

then a max-density condition with parameters δ and r_0 according to Definition 1 holds.

Proof. We first note that Δ is well-defined and positive because from $\delta r_0^2 > 1$ if follows that $\sqrt{\delta} - r_0^{-1} > 0$. We must show that for any circle C with radius $r \geq r_0$ less than δr^2 points can be fit into C, if the pair-wise distances between the points are all at least Δ. The latter is equivalent to stating that if we draw a circle with radius $\frac{\Delta}{2}$ around all measurement points within C, then none of these small circles may intersect with each other.

Observe that the centers of all small circles must lie within C. Therefore, all the small circles fully lie within a circle C^+ with the same center as C and radius $r + \frac{\Delta}{2}$. Thus, the total area of the small circles (each covering an area of $\pi(\Delta/2)^2$) is less than the area of C^+, which is $\pi(r + \Delta/2)^2$. (It is strictly less because C^+ can impossibly be fully covered by small circles.) Therefore, for the total number n of small circles (i. e., the total number of measurement points within C) it holds that

$$n < \frac{\pi\left(r + (\Delta/2)^2\right)}{\pi(\Delta/2)^2} = \frac{r^2}{(\Delta/2)^2} + \frac{2r}{\Delta/2} + 1 = r^2\left(\sqrt{\delta} - \frac{1}{r_0}\right)^2 + 2r\left(\sqrt{\delta} - \frac{1}{r_0}\right) + 1. \tag{5.6}$$

Because $r \geq r_0$, n is thus bounded above by

$$r^2\left(\sqrt{\delta} - \frac{1}{r}\right)^2 + 2r\left(\sqrt{\delta} - \frac{1}{r}\right) + 1 = r^2\delta. \tag{5.7}$$

Consequently, the max-density condition holds. ■

Lemma 2. *On the boundary of a circle with radius $r \geq \frac{\Delta}{2}$, at least $\lfloor \frac{4r}{\Delta} \rfloor$ points can be positioned such that for each pair of points their distance is at least Δ.*

Proof. We show that when $\lfloor \frac{4r}{\Delta} \rfloor$ points are evenly distributed on the boundary of a circle with radius $r \geq \frac{\Delta}{2}$, the distance between two neighboring points is at least Δ. The angle α between two neighboring such points on the circle is

$$\alpha = \frac{2\pi}{\lfloor \frac{4r}{\Delta} \rfloor} \geq \frac{2\pi}{\frac{4r}{\Delta}} = \frac{\pi}{2}\frac{\Delta}{r}. \tag{5.8}$$

The distance μ between the points is the base of an isosceles triangle with arm length r and angle α. Hence,

$$\mu = 2r\sin\left(\frac{\alpha}{2}\right). \tag{5.9}$$

Note that $\forall x \in [0, \pi/2]$ it holds that $\sin(x) \geq \frac{2x}{\pi}$, and also note that $\alpha < \pi$ (because $r \geq \frac{\Delta}{2}$ and thus $\lfloor \frac{4r}{\Delta} \rfloor \geq 2$). Therefore,

$$\mu \geq 2r\frac{2\alpha}{\pi} \geq 2r\frac{2}{\pi}\frac{\pi\Delta}{4r} = \Delta. \tag{5.10}$$

From the fact that the distance between each pair of neighboring points is at least Δ, it easily follows that all pair-wise distances are at least Δ. ■

In the following, we will assume that the parameters δ and r_0 from the max-density condition as well as the parameters ξ and r_1 from the bandwidth limit assumption are fixed and given. Let furthermore Δ in the following denote the quantity $\frac{2}{\sqrt{\delta - r_0^{-1}}}$, as in Lemma 1.

We will now show that any bandwidth profile must be in $o(1/d^2)$, or otherwise there is a possible constellation of measurement points and interests such that a max-density condition holds, but not all interests i can be served with bandwidth $b(\|i\|)$. We prove this by contradiction. Therefore, let from now on b be a bandwidth profile for which

$$b(d) \notin o\left(\frac{1}{d^2}\right). \tag{5.11}$$

We will proceed in three steps towards a proof. We first construct a set of measurement points M^* and set of interests $\mathcal{I}^*$ based on the parameters δ, r_0 from the max-density condition. We then show that for this construction the max-density condition holds. Finally, we prove that serving all interests $i \in \mathcal{I}^*$ with bandwidth at least $b(\|i\|)$ is infeasible.

Note that $b \notin o(1/d^2)$ means that there is a constant $c > 0$ such that

$$\forall d_0 > 0 : \exists d > d_0 : b(d) \geq \frac{c}{d^2}. \tag{5.12}$$

Let in the following c be such a constant.

Definition 5. *Let* $k_0 := \max\{r_1, r_0, \Delta\}$.

For all $i \in \mathbb{N}_{>0}$ *let* $k_i \in \mathbb{R}$ *be a value for which*

$$k_i > 8k_{i-1} + \Delta \qquad \text{and} \qquad b(k_i) \geq \frac{c}{k_i^2}. \tag{5.13}$$

Such k_i exists for all k_{i-1} because $b(d) \notin o(1/d^2)$ (refer to the remark above for $d_0 = 8k_{i-1} + \Delta$ and $d = k_i$).

Based on the sequence $(k_i)_{i\in\mathbb{N}}$ we can now construct M^* and $\mathcal{I}^*$ as follows.

Definition 6. *Let M^* be a set of measurement points defined as follows.*

We first construct a sequence of primary circles. *The primary circles are all centered at the origin. The i-th primary circle, denoted by $C_{i,0}$, has radius k_i.*

For all $i \in \mathbb{N}_{>0}$ we construct further circles, called secondary circles. *The secondary circles, too, are centered at the origin. Between $C_{i-1,0}$ and $C_{i,0}$, there are $w_i - 1$* secondary circles, *denoted by $C_{i,1}, \ldots, C_{i,w_i-1}$, where*

$$w_i = \left\lfloor \frac{k_i - k_{i-1}}{\Delta} \right\rfloor . \tag{5.14}$$

The radius of $C_{i,j}$ is $k_i - j\Delta$.

Figure 5.1 visualizes the construction with primary circles and secondary circles (dashed).

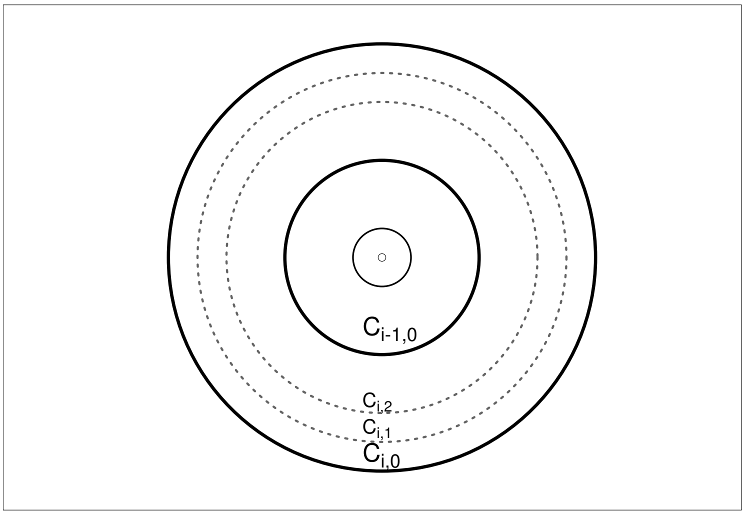


Figure 5.1: Primary and secondary circles in the construction of M^*.

According to Lemma 2,

$$\left\lfloor \frac{4(k_i - j\Delta)}{\Delta} \right\rfloor \tag{5.15}$$

points can be positioned on $C_{i,j}$, such that they all have a pair-wise distance of at least Δ. M^* consists of such points on all circles $C_{i,j}$. The exact placement of the points on the circle is not relevant in the following, as long as the pair-wise distance is at least Δ; for *instance, one point on $C_{i,j}$ could be positioned at coordinates $(0, k_i - j\Delta)$, with the rest of the points uniformly spaced over the circle boundary.*

Let the i-th zone*, Z_i, be the subset of M^* that contains all measurement points that reside on the circles $C_{i,0}, \dots, C_{i,w_i-1}$ (i. e., it comprises the primary circle $C_{i,0}$ and all secondary circles between $C_{i-1,0}$ and $C_{i,0}$). Let z_i denote the number of measurement points in zone Z_i.*

Finally, let

$$\mathcal{I}^* = \{((0,0), m) \mid m \in M^*\}.$$

We are now ready to prove that a max-density condition with parameters δ and r_0 holds for our construction.

Theorem 1. *A max-density condition with parameters δ, r_0 holds for M^*.*

Proof. We will show that for any two measurement points $m_1, m_2 \in M^*, m_1 \neq m_2$ the distance $\|m_1 - m_2\|$ is at least Δ. First, note that according to Definition 6 the difference between the radii of any two circles is at least Δ. Therefore, if m_1 and m_2 reside on different circles, their distance must be at least Δ. If m_1 and m_2 are located on the same circle, however, their distance is also at least Δ by construction.

Consequently, the distance between any pair of measurement points is at least Δ. Thus, by Lemma 1, the max-density condition holds. ∎

We now turn towards the number of measurement points within the individual zones and make the following observation. First, however, we show a simple lemma which will be of great help in the proof.

Lemma 3. *For all $k \in \mathbb{N}, k \geq 1$ and all $x \geq k$ it holds that*

$$\lfloor x \rfloor > \frac{k}{k+1} x. \tag{5.16}$$

Proof. For $x \in \mathbb{N}$ the assertion trivially holds. Thus, we focus on the case $x \notin \mathbb{N}$. Let $y := x - \lfloor x \rfloor$. Since $k \in \mathbb{N}$ and $x \geq k$ we also have that $\lfloor x \rfloor \geq k$. Because $0 < y < 1$ we get

$$\begin{aligned} \frac{\lfloor x \rfloor}{y} > k \quad &\Rightarrow \quad \lfloor x \rfloor > ky \quad \Rightarrow \quad \lfloor x \rfloor > k(x - \lfloor x \rfloor) \\ \Rightarrow \quad (k+1)\lfloor x \rfloor > kx \quad &\Rightarrow \quad \lfloor x \rfloor > \frac{k}{k+1} x. \end{aligned} \tag{5.17}$$

∎

We can now establish a lower bound on the number of measurement points in a zone Z_i.

Lemma 4. *Let $i \in \mathbb{N}_{>0}$. For the size z_i of zone Z_i it holds that*

$$z_i > \frac{k_i^2}{2\Delta^2}. \tag{5.18}$$

Proof. According to the definition of M^* above, the total number of measurement points in zone Z_i is

$$z_i = \sum_{j=0}^{w_i-1} \left\lfloor \frac{4(k_i - j\Delta)}{\Delta} \right\rfloor \tag{5.19}$$

Since $\frac{4(k_i - j\Delta)}{\Delta} > 1$ for any $j < w_i$, we can apply Lemma 3 and get

$$\begin{aligned} z_i &> \sum_{j=0}^{w_i-1} \frac{2(k_i - j\Delta)}{\Delta} \\ &= 2\sum_{j=0}^{w_i-1} \frac{k_i}{\Delta} - 2\sum_{j=0}^{w_i-1} j \\ &= 2\left\lfloor \frac{k_i - k_{i-1}}{\Delta} \right\rfloor \frac{k_i}{\Delta} - 2\frac{(w_i-1)w_i}{2}. \end{aligned} \tag{5.20}$$

Since per definition $k_i > 8k_{i-1}$ and $\forall i \in \mathbb{N} : k_i \geq \Delta$ it holds that

$$\frac{k_i - k_{i-1}}{\Delta} > 7. \tag{5.21}$$

We may thus again apply Lemma 3 and obtain

$$z_i > \frac{7(k_i^2 - k_i k_{i-1})}{4\Delta^2} - (w_i - 1)w_i. \tag{5.22}$$

Recall that the number of circles in zone Z_i is

$$w_i = \left\lfloor \frac{k_i - k_{i-1}}{\Delta} \right\rfloor \tag{5.23}$$

and therefore

$$1 \leq w_i < \frac{k_i}{\Delta}. \tag{5.24}$$

Thus,

$$(w_i - 1)w_i < \frac{k_i^2}{\Delta^2} \tag{5.25}$$

and we arrive at

$$\begin{aligned} z_i &> \frac{7(k_i^2 - k_i k_{i-1})}{4\Delta^2} - \frac{k_i^2}{\Delta^2} \\ &= \frac{3}{4}\frac{k_i^2}{\Delta^2} - \frac{7}{4}\frac{k_i k_{i-1}}{\Delta^2} \\ &= \frac{1}{2}\frac{k_i^2}{\Delta^2} + \left(\frac{1}{4}\frac{k_i^2}{\Delta^2} - \frac{7}{4}\frac{k_i k_{i-1}}{\Delta^2}\right) \\ &= \frac{1}{2}\frac{k_i^2}{\Delta^2} + \left(\frac{(k_i - 7k_{i-1})k_i}{4\Delta^2}\right) \end{aligned} \tag{5.26}$$

Since per definition $k_i > 7k_{i-1}$, the term in parentheses is positive and thus

$$z_i > \frac{k_i^2}{2\Delta^2}. \tag{5.27}$$

This is the assertion. ■

Theorem 2. *Let b be an arbitrary bandwidth profile for which $b(d) \notin o(1/d^2)$. Then, for measurement points M^* and interest $\mathcal{I}^*$ as defined above, not all interests $i \in \mathcal{I}^*$ can be served with bandwidth at least $b(\|i\|)$.*

Proof. Let M^* and I^* be defined like above. Consider a circle C^* centered around 0 with radius r_1. Note that all circles in the definition of M^* have a radius larger than r_1 (since $k_0 \geq r_1$); thus, all points in M^* are located outside C^*. Since for each point $m \in M^*$ there is an interest $i_m = ((0,0), m) \in \mathcal{I}^*$, information about m must be transported into C^* at least with bandwidth $b(\|i_m\|)$. Note that $\|i_m\|$ is equal to the radius of the measurement point circle (according to the definition of M^*) on which m is located.

Now observe that for each measurement point m in zone Z_i, the bandwidth that must be spent for the point when delivering information for interest i_m is bounded below as follows

$$b(\|i_m\|) \geq b(k_i), \tag{5.28}$$

because b is monotonically decreasing per definition and $\|i_m\| \leq k_i$.

Therefore, the total bandwidth B that must be transported into C^* to serve all interests according to b is—by summation over the zones—bounded below by

$$B \geq \sum_{i=1}^{\infty} b(k_i) z_i \geq \sum_{i=1}^{\infty} \frac{c}{k_i^2} z_i. \tag{5.29}$$

According to Lemma 4, we thus have

$$B \geq \sum_{i=1}^{\infty} \frac{c}{k_i^2} \left(\frac{k_i^2}{2\Delta^2} \right) = \frac{c}{2\Delta^2} \sum_{i=1}^{\infty} 1. \tag{5.30}$$

This sum obviously does not converge, therefore infinite bandwidth into the circle C^* were necessary to serve all interests. Since the bandwidth available for transmissions into C^* is finite (bounded above by ξr_1 according to Assumption 1), the assertion holds. ■

5.3 Conclusion

In this chapter we have analyzed the fundamental limit of information dissemination. Instead of spreading continuously updated information in the network which would rapidly congest it, we focused on techniques that summarize and aggregate data dependent to their covered distance. In order to determine the minimal "amount" of aggregation necessary we assumed that measured information is aggregated and transported in a network with limited capacity. We showed theoretically that an aggregation scheme's bandwidth needs to be in $o(1/d^2)$, where d is the covered distance, in order to be scalable. This means that it is not feasible to spread information in the network if the amount of data is reduced quadratically or even less with the distance.

Chapter 6

Data Aggregation and Roadside Unit Placement for a VANET Traffic Information System

In the last two chapters we recognized that information dissemination applications based on VANETs face two key challenges: a limited network capacity shared by all cars and—at least initially—a highly partitioned network limiting the speed of data dissemination. In this chapter we tackle these problems in the context of a cooperative traffic information system, where all participating vehicles gather data on the current traffic situation. This information is then distributed so that other cars may use it for improved route planning.

As we have seen in Chapter 5, it is not possible to distribute all data to all equipped vehicles. This would quickly exceed the available bandwidth. Instead, we propose to aggregate it in a hierarchical fashion: the farther away a region is, the coarser will be the information on its traffic situation. The general idea of hierarchical aggregation is not new. However, existing hierarchical aggregation schemes focus on combining data from geographical regions, with aggregates representing averages or extremal values within these regions. This may be fine for aggregating information like the availability of parking places. It is useful to have an aggregate describing the total number of available parking places within some region. For a traffic information system, however, it is not sufficient to know geographical averages or extremal values. We therefore propose a hierarchical aggregation scheme for travel times in road networks. Essentially, we use coarser and coarser approximations of the road network to summarize travel times in regions that are farther and farther away.

The aggregation scheme allows to deal with network capacity limits by summarizing the collected data. But we know that VANETs also suffer from very limited connectivity and from limited information propagation speed, especially during the early rollout phase. However, we have discussed in Chapter 4 to improve information dissemination by adding comparatively inexpensive infrastructure at some locations in a city. These stationary supporting units (SSUs) exchange information with cars passing by and, using a backbone network, with each other, thereby delivering information rapidly to more distant regions. However, while we have shown that even a very limited number of SSUs can largely improve the dissemination process, it is not yet clear where they should be located.

Therefore, as the second major contribution of this chapter, we present an optimization methodology to find a good placement for supporting units in the context of a VANET-based traffic information system. The presented approach makes use of a genetic algorithm in order to maximize the travel time savings of cars in a city environment.

The remainder of this chapter is structured as follows. Related work is reviewed in Section 6.1. In Section 6.2 we introduce our scheme for aggregating travel time information in city environments. We show in Section 6.3 how a combination of good SSU locations for a traffic information system on the basis of this aggregation scheme can be found. We present and discuss the results of a simulative evaluation in a VANET city scenario in Section 6.4. Finally, we conclude this chapter with a summary in Section 6.5.

6.1 Related Work

Developing an advanced traffic information system (ATIS) that makes use of current information about the condition of streets is complex in different tasks. The most important goal to be reached is spreading the information to vehicles that are in interest for it. Only if this is done in a reliable manner, a route calculation algorithm can be used. In the following paragraphs we want to review some studies in the context of these calculation methods.

TraffCon is a hybrid approach consisting of a wireless (mesh) network to disseminate road condition information to a server based navigation system presented in [CM08]. The authors propose to use vehicle based making of observations that will be sent in combination with a desired destination to a routing server. This server then calculates the current fastest route to the destination as requested. The calculation is based on a fitness function that aims to decrease the emission of CO_2 as well as congestion at one particular road. The route is then sent back to the vehicle by using the wireless network.

Building a hierarchy in order to facilitate the solving of a problem is not specific to any application. Thus, similar strategies are useful for different challenges. In different studies [BFSS07, SS06, NBB$^+$08, DW07] the respective authors present approaches that also deal with navigation systems. But their main focus is on improving the calculation process of favorable routes in such a system. By using different layers the calculation process can be split into several smaller calculation processes. The connection points between these layers are called landmarks. The idea behind this is that a car will first reach a distinct landmark on a lower layer and then will follow a route on the next higher layer and so on. This is a naturally implementation of existing street types.

The navigation system proposed in this thesis also makes use of landmarks. Nevertheless, the focus is orthogonal to the one above. Landmarks are mainly used for the creation of aggregates. At a first glance this seems to be very similar. However, in this thesis the focus is not on calculating routes, we consider the creation of aggregates to disseminate information to further regions. Indeed, both approaches could be easily combined. This would on the one hand allow to disseminate information on traffic situations. On the other hand these aggregates could then be used by the above mentioned approaches to calculate appropriate routes in a very efficient manner[1].

6.2 Aggregation

The basic idea of our aggregation scheme is as follows: we define landmarks on multiple levels of a hierarchy in the road network. At the highest level these are junctions of the main roads or highways. Lower levels include all higher level landmarks plus more and more intersections of smaller streets. The lowest level is a representation of the full road network. Cars passing a road segment can make an observation of the current travel time between two neighboring landmarks. This information is distributed within the closer surrounding. It is used by cars to calculate travel times between landmarks of the next higher level, thereby summarizing the travel times in the area. This coarser picture on the travel times is distributed in a larger area than the observations of individual cars. It is also used to calculate the travel times between landmarks of the next higher level of the hierarchy, and so on.

These aggregation steps are performed by the cars themselves, in a completely decentralized fashion, whenever information becomes locally available that is a suitable basis for forming an aggregate.

[1] In our implementation we use an A* shortest path algorithm [HNR68, HNR72] for the calculation of shortest paths.

6.2.1 Aggregation on a Single Level

Figure 6.1 depicts an example situation for a single hierarchy level. The travel time between the landmarks *Eiffel Tower* and *Arc de Triomphe* is determined. These two high-level landmarks are connected via a number of possible routes over landmarks on the next lower level, here indicated by circles. The travel times between these lower-level landmarks are known, either from direct observations or from received or previously calculated lower-level aggregates. The aggregated travel time from Eiffel Tower to Arc de Triomphe is the travel time along the minimal travel time route between these two points. Essentially, this compresses all information on all possible paths between two landmarks to a "virtual" link between both landmarks.

This approach can be stated more formally as follows. The road network can be seen as a directed graph $G(E,V)$ consisting of junctions $v \in V$ and street segments $e \in E \subseteq V^2$ connecting these junctions. The segments are rated with a weight $w(v_1, v_2)$ corresponding to the current travel time. Some junctions are distinguished as landmarks $l \in L, L \subseteq V$. A route $r(A,B)$ between two landmarks A and B is a sequence of junctions $(v_1,\ldots,v_n)$ such that $v_1 = A$, $v_n = B$ and all pairs of consecutive junctions are connected by a street segment, i. e., for all $i = 1,\ldots,n-1$ there exists $(v_i, v_{i+1}) \in E$. The cost of this route is

$$\|(v_1,\ldots,v_n)\| := \sum_{i=1}^{n-1} w(v_i, v_{i+1}). \tag{6.1}$$

Let $R(A,B)$ denote the set of all possible routes from A to B. We can then define the fastest route $r^*(A,B)$ as follows:

$$r^*(A,B) := \operatorname*{argmin}_{r \in R(A,B)} \|r\|. \tag{6.2}$$

$\|r^*(A,B)\|$ is used as the travel time between landmarks A and B on the next higher level. I. e., the operation performed when calculating an aggregate is to determine $\|r^*(A,B)\|$ based on lower-hierarchy travel times. Any standard routing algorithm may be used in order to calculate $\|r^*(A,B)\|$. Note that it is *not* relevant which route actually achieves this travel time. While the car is further away from both landmarks, the relevant information is that it is *possible* to travel from A to B within the given time. When it comes closer to the respective area, it will receive the locally available, more detailed information. This information can then be used for routing.

Aggregated travel times should not be calculated for each pair of landmarks. First, this would result in a number of aggregates that grows like $O(n^2)$ with the number of landmarks. More importantly, however, travel times aggregates between landmarks that are very far away from each other do also not contribute much additional information: there

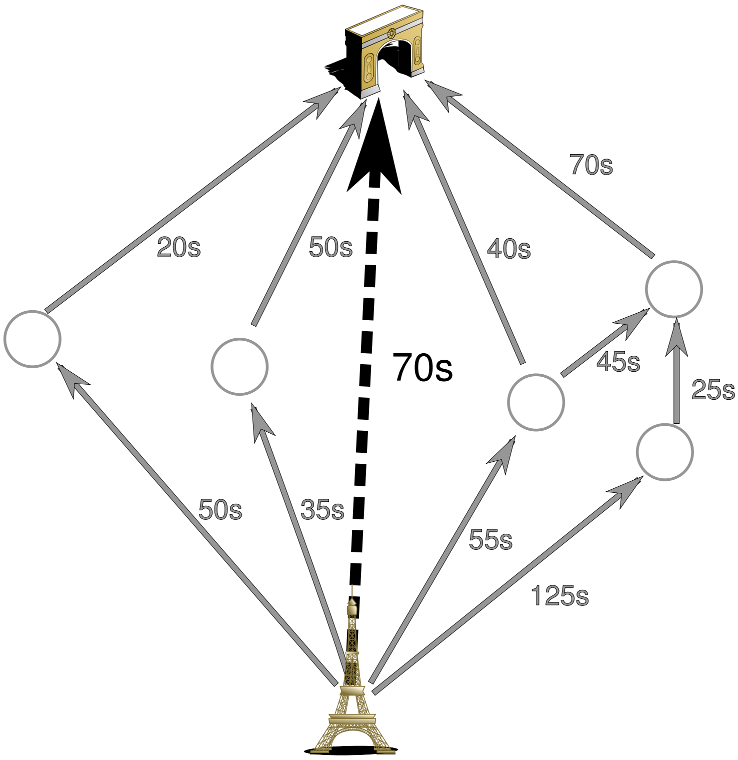


Figure 6.1: Landmark aggregation.

will be many other landmarks "in between", and a sequence of aggregates over those is likely to be a good approximation of the travel time between the distant pair of landmarks. Hence, only such pairs should be considered that are not too far apart. It is either possible to use a fixed criterion, like a maximum beeline distance of landmark pairs for which aggregates are formed on a given hierarchy level, or to mark the landmark pairs explicitly in the map data, choosing them such that a good approximation of the underlying road network is maintained. In our evaluation here, we follow the latter approach.

6.2.2 Judging the Quality of Information

Before calculating the aggregated travel time between two landmarks and passing it on to other cars, a car needs to be able to judge whether its knowledge about the current traffic situation suffices for a good estimate. From a very general perspective, a very large number of road segments could lie on a possible route between two landmarks. It is therefore necessary to determine which road segments are likely to be relevant for the travel time estimate. An aggregate may be formed if information on these relevant road segments is locally available.

In order to define the set of relevant road segments, we look at what we call the *standard travel times* along the road segments. These travel times are hard-coded in the road map data and represent reasonable expectations of the travel times, as they are currently used for non-dynamic road navigation systems. One can easily calculate the *optimal standard route* $r^*_{\text{std}}(A,B)$ from A to B on the basis of this static data—this is essentially what current navigation systems do. For any route r, it is also easily possible to calculate the standard travel time $\|r\|_{\text{std}}$.

We then choose a threshold $\theta > 1$. We define the set $\mathcal{R}(A,B)$ of relevant road segments between two landmarks A and B to encompass all road segments that lie on a route for which the standard travel time is at most by a factor of θ longer than the optimal standard travel time. I. e., a road segment e is in $\mathcal{R}(A,B)$ if and only if there exists a route $(v_1,\ldots,v_n)$ from A to B such that e is part of that route and

$$\|(v_1,\ldots,v_n)\|_{\text{std}} \leq \theta \cdot \|r^*_{\text{std}}(A,B)\|_{\text{std}}. \tag{6.3}$$

Since this criterion is based only on the (static) standard travel times, the set of relevant road segments does not depend on the current traffic situation or on a car's current knowledge. We allow the calculation of an aggregated travel time from A to B if information on the road segments in $\mathcal{R}(A,B)$ is available.

6.2.3 Hierarchical Aggregation

In order to perform hierarchical aggregation, landmarks are assigned a level in a hierarchy. The number of landmarks corresponds to the results of Chapter 5. Landmarks of a higher level are also members of all lower levels. More formally spoken, for a set of landmarks L_i of an aggregation level i:

$$L_i \subset L_{i-1} \subseteq V, \quad i > 1. \tag{6.4}$$

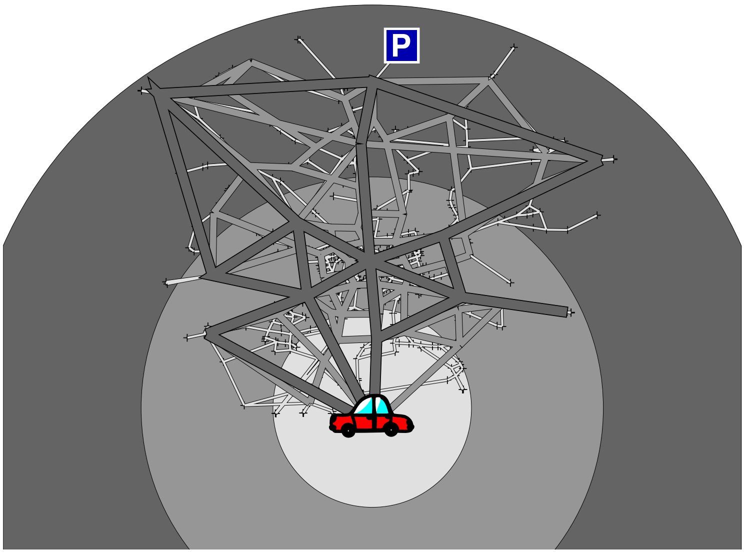
To form an aggregate on hierarchy level i, the aggregates of level $i-1$ are used in the very same way as individual observations of cars are used on the first level. Thus, the landmarks on level $i-1$ are used like the junctions in the discussion above, and the aggregated travel times between them take on the role of the travel times along individual street segments.

The area in which individual observations and aggregates are distributed is, by design, limited based on their level in the hierarchy. Individual observations are distributed in a very limited range whereas the highest level aggregates are distributed in the whole network.

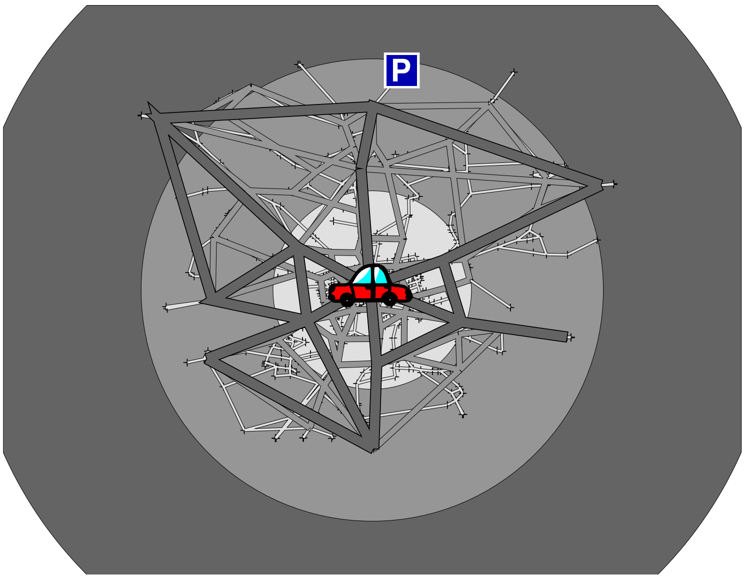
In Figure 6.2(a), this aggregation mechanism is depicted. The driver of the car located in the south of the scenario intends to travel towards the north. Three hierarchy levels can be seen here. The circles around the car indicate the regions from which detailed, fine-grained level 0 information, slightly aggregated level 1 information, and more coarsely aggregated level 2 information is available to this car. As the car travels towards its destination, these regions shift as shown in Figure 6.2(b), as additional information is received.

The destination of a trip will not always be a high-level landmark position. Nevertheless, the aggregated information can of course be used for route planning. In order to do so, a navigation system would "fill up" the missing information between the final destination and close-by landmarks by using the standard travel times which are hardcoded in the map data.

This is reasonable, because a final decision on the last part of the route is not yet required at this stage—it is sufficient if a good choice for the immediately upcoming routing decisions can be made. While the car approaches its destination, the route can be updated and refined as more detailed information becomes available.



(a) Starting point.



(b) During the trip.

Figure 6.2: Hierarchy based navigation.

6.3 Placement of Supporting Units

During the rollout of car-to-car technology the equipment density of cars participating in a VANET will be low. This makes timely information dissemination very difficult. It is the second key problem that has to be solved for a VANET-based traffic information system. In Section 4.4 we have proposed to make use of infrastructure devices—*stationary supporting units (SSU)* or simply *supporting units*—to improve dissemination performance. SSUs use the same radio technology and essentially the same application as equipped vehicles. They are able to receive observations and aggregates from passing-by cars. They also send beacons and thereby hand over their knowledge to cars. The central benefit of SSUs, however, is achieved by connecting them via a backbone network, allowing them to exchange information. This can bring up-to-date knowledge to distant network regions in very short time.

A very limited number of SSUs is sufficient for substantial benefits. But nevertheless SSUs incur deployment and maintenance costs. Hence, the question arises how to achieve good performance with as few SSUs as possible, or—putting it the other way around—at which positions in the road network of a given city a given number of SSUs should be located in order to achieve high benefit. This question is closely related to the employed aggregation scheme, because both in conjunction determine which information will be able to arrive at which location and which point in time.

In the following paragraphs, we propose a way to position SSUs such that their application level benefit is maximized. To this end we define an application level metric for a traffic information system. This metric reflects the travel time saved by using the application. Then we show how a genetic algorithm (GA) can be used to optimize the placement with respect to this metric.

6.3.1 Optimizing Supporting Units Placements

In a city, there are typically many possible positions for SSUs. Given a set of potential SSU locations and a number of SSUs to be placed, our approach aims to identify the optimal subset of locations. For a given SSU positioning, it is theoretically conceivable to run a simulation (using, e. g., an integrated simulation environment that models both car movement and network traffic) and to measure the achieved travel time saving. But even with a moderate number of possible locations and SSUs, the number of possible combinations is overwhelming. If there are 100 potential locations for 10 SSUs, there are $1.73 \cdot 10^{13}$ possible placements. With 30 SSUs, there are $2.9 \cdot 10^{25}$ possibilities. Therefore it is obviously not feasible to assess and compare all placements. Identifying the optimal subset of SSU locations actually turns out to be a very difficult optimization problem. Here, we use genetic algorithms in order to find a good approximation.

Basically, genetic algorithms start off with a random set of "individuals" (SSU placements), assess their "fitness" (achieved travel time savings) , and then generate a new "generation" of individuals by combining features of the "fittest". This approach has been applied to a broad range of problems, and often yields excellent results. A more detailed description can be found in [Gol89, SD07]. Nevertheless, the computational effort for assessing SSU placements remains significant. Typically, at least several dozens of generations are necessary, each with many individuals. For each of these individuals the fitness—i. e., the objective function—needs to be calculated.

A fully-fledged simulation would model car movement, the network (i. e., radio propagation, medium access, exchanged beacons etc.), and the application (i. e., the contents of the beacons) in parallel. This is computationally very expensive (with standard traffic and network simulators at least a few hours of computation time per individual), and is thus still not possible for all these individuals within reasonable time. Thus, some approximations must be found, which significantly speed up the simulation, but still capture the relevant effects in sufficient detail. We propose such a method to obtain an estimate of travel time savings within a comparatively short time (typically 1–3 minutes). This is made possible by de-coupling the application-layer simulation from the lower layers.

6.3.2 Estimating Travel Time Savings

Dissemination takes place by periodic beaconing by both cars and SSUs. Though other approaches are generally conceivable, in the majority of schemes and also in the mechanisms employed here, the transmission of these beacons does not depend on their contents or on the sender's knowledge. PHY and MAC effects do also not depend on the data within the packets. The points in time at which the network nodes transmit and the set of nodes receiving each transmission do therefore *not* depend on the transmitted data.

Therefore, we may simulate the network traffic *independently* from the application. In practice, we use VISSIM for the simulation of car movements in a city, and ns-2 for the simulation of periodic beacons issued by all cars and SSUs; other traffic and/or network simulators could be used without generally affecting the proposed methodology. This simulation step yields a log file in which all beacons are recorded with their respective receiver set, along with the car positions.

We may then—subsequently—run a separate *application simulator* that reads this log file, keeps track of the knowledge base of all nodes, performs aggregation as specified in the previous section, decides about the data contained in each of the beacons, and respectively updates the knowledge bases of all receivers of the beacon. This does not allow to model the impact of changing knowledge on the behavior of the cars (in particular, their route will not depend on their knowledge). But unless the large-scale traffic flow is significantly impacted by use of the application (implying that a very large fraction of cars use it), the effects will be negligible.

The so far described approach is able to separate traffic and network simulation from the simulation of the application. But the network simulation processes beacons sent by stationary supporting units and therefore still depends on the SSU positioning. This is where one of the central approximations made by our method comes into play: we simulate SSUs at *all* possible SSU locations in the network simulator, and let *all* of them transmit periodic beacons. In the application simulator, we then simply ignore beacons from "non-existing" SSUs; the respective beacons are not considered when updating the knowledge bases.

The number of SSU locations is small in relation to the number of cars. So, the vast majority of network traffic is caused by cars, and occasional beacons sent at unoccupied SSU locations have only negligible effects. But this simplification allows us to re-use a traffic and network simulation log file for application simulations of any arbitrary set of SSU locations. A set of such log files may be precomputed. A flow-chart outlining the toolchain is depicted in Figure 6.3.

We can now draw samples of estimated travel time savings in the following way. The application simulator assigns travel times to all road segments. Our implementation picks a pre-configured fraction γ of the road segments and assigns them a travel time that is ϕ times higher than their respective standard travel time; the travel time of all other segments is set equal to the standard travel time. If more sophisticated models for the travel times become available, they can of course easily be plugged in.

We choose a car at a specific point in time. The car's position at this time is known, as well as its (not necessarily perfect) knowledge about the current traffic situation. We also choose a random destination for the car. We then calculate the optimal route r^* from the car's current position to this destination, based on this car's knowledge. We also calculate the optimal standard route r^*_{std} based on standard travel times as introduced in Section 6.2.2. This is the route a standard navigation system would choose.

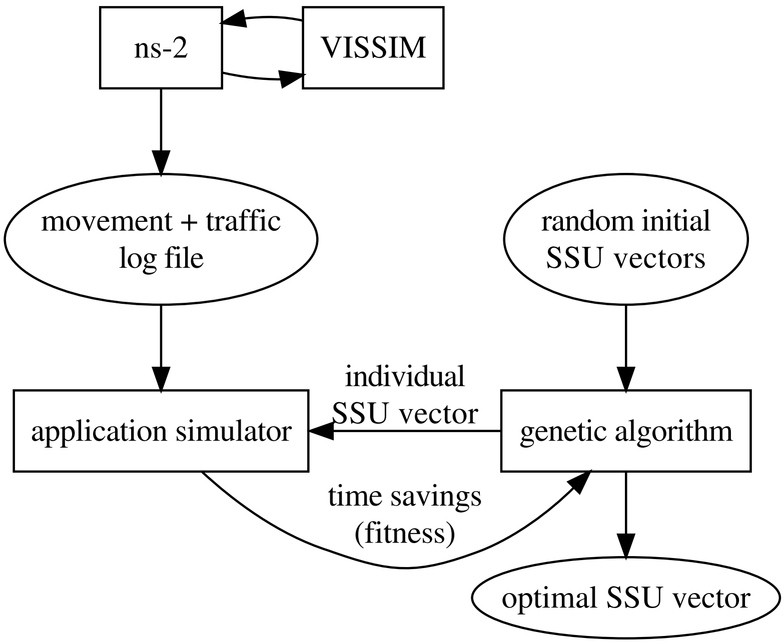


Figure 6.3: Toolchain used for SSU placement optimization.

For both routes, we calculate the travel times based on the true current travel times as set by the application simulator. The ratio of these two travel times is used as the estimated travel time saving. Note that the car's current knowledge is typically not perfect. It will virtually always deviate from the current traffic situation to some extent (e. g., because the situation changes over time). The dynamic route might therefore even be worse than the standard route. The travel time benefit is thus highly dependent upon the dissemination performance: it will be high if up-to-date information relevant for the route calculation is known by the car. The mean travel time savings over many such samples can be used as a metric for the application benefit.

Of course, this approach is an approximation in many regards. The simulation methodology makes, as already discussed, a number of simplifying assumptions. Furthermore, for example, cars do not actually follow their dynamically chosen routes in the simulation, and thus information potentially obtained during their travel and subsequent route adjustments are not taken into account. The central benefit of the method, however, is that it allows to obtain a good estimate with limited computational effort. It is therefore usable as an objective function for the optimization of the SSU placement.

6.3.3 Genetic Algorithm

Based on the method for calculating the objective function for a given SSU placement, a genetic algorithm (GA) can be used to actually find a good SSU placement. To this end we need to express the set of occupied SSU locations as an "individual". There is a rather natural representation for this purpose: a bit string, where each bit position stands for one SSU location. A bit is set to one if and only if there is a stationary supporting unit at this position. An example representation is depicted in Figure 6.4. There are ten possible positions of SSUs, four of them are used.

SSU_0	SSU_1	SSU_2	SSU_3	SSU_4	SSU_5	SSU_6	SSU_7	SSU_8	SSU_9
1	0	1	1	0	0	1	0	0	0

Figure 6.4: The genetic representation of supporting units.

Selection and Recombination

For each generation, the fitness of each individual is computed. We do so using the methodology described above. Then the sum of the fitness values of all individuals in the generation is calculated. The selection of parents for the next generation is then performed by randomly selecting individuals based on their relative fitness, i. e., their own fitness divided by the sum.

For a given, fixed number of SSUs, we want to optimize over all bit vectors with the respective number of bits set. Therefore, when combining two parent individuals, it needs to be ensured that the newly created individual still has the same number of bits set. Standard recombination techniques like uniform or multi-point crossover cannot guarantee this. We propose to use the following recombination technique: in the child's bit vector, we first set all bit positions that are set in *both* parents (i. e., we start with the bit-wise AND of the parents). We then "fill up" the bit vector by setting additional bits. We choose these additional bits randomly from all bits set in exactly *one* of the parents (i. e., from those bits enabled in their bit-wise XOR).

Mutation

After creating children individuals, it is vital for genetic algorithms that "mutations" happen. They avoid that the algorithm gets trapped in a local optimum. Mutation is done by randomly flipping some (few, here we use 0.4 %) of the bit positions. Again we need to make sure that the number of set bits remains constant. We therefore chose to mutate by simply exchanging the state of two bits in the bit vector.

Parallelization

In our implementation, we use a Multi-Population Genetic Algorithm (MPGA) with two separate populations. It has been shown that this leads to faster convergence properties for genetic algorithms; for a survey see [CP95]. After each generation it is possible that an individual migrates to the other population.

Search Termination

We stop the genetic algorithm if we arrive at homogeneous populations, i. e., if one specific set of SSU locations dominates the populations. If no homogenization occurs, we stop the algorithm after a fixed number of generations.

The operation of the genetic algorithm is summarized in Figure 6.5. Whenever one iteration (i. e., one generation of individuals) is finished, the individuals of the old population are replaced by newly recombined and mutated ones. In order not to lose the information of the so far overall best individual, the memorized chromosome is also included into the new population. In the last step of each iteration, the algorithm randomly chooses an individual from each population. These individuals are then moved to the other population. After this final step the genetic algorithm proceeds with the next generation.

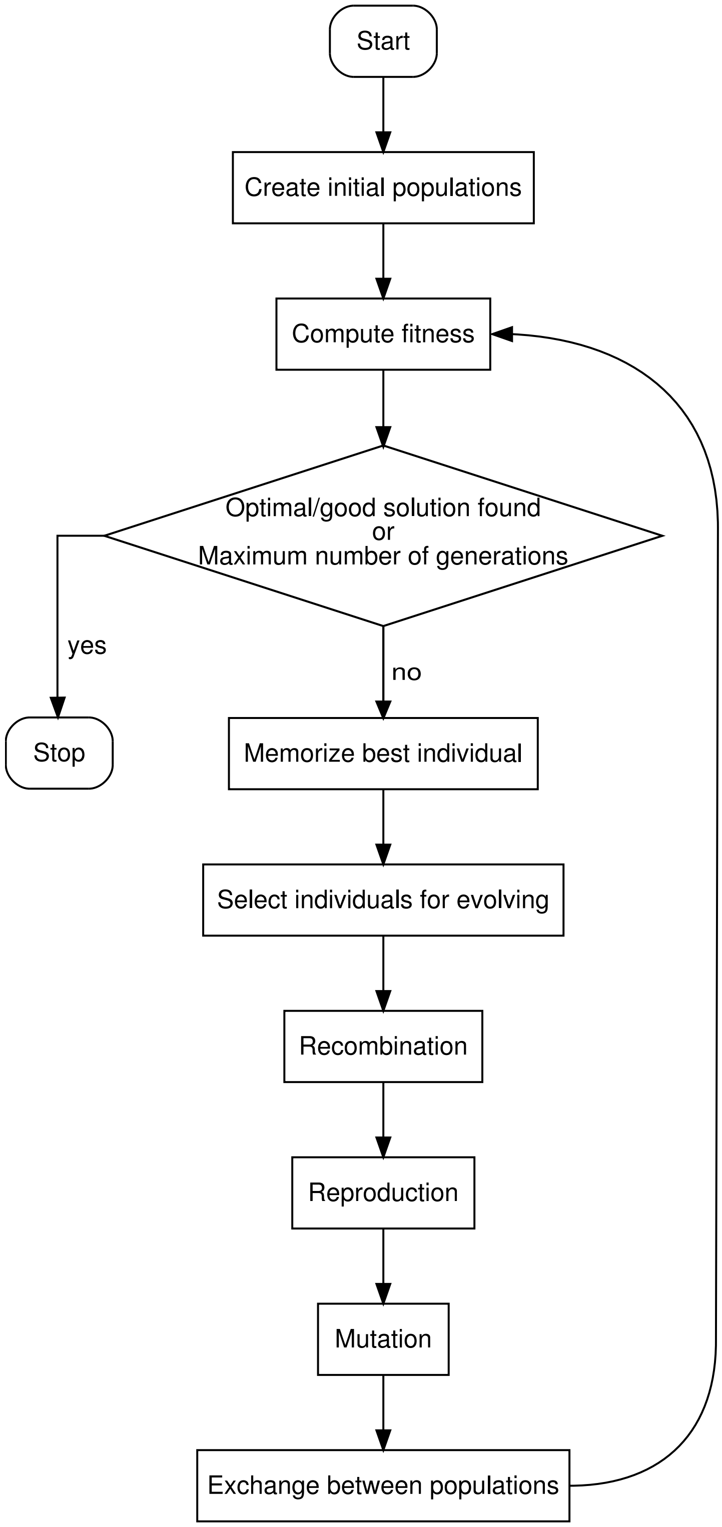


Figure 6.5: The procedure of a GA based optimization approach.

6.4 Evaluation

In order to evaluate the aggregation scheme and SSU placement methodology outlined above, we apply it to a specific VANET city scenario model and analyze aspects of both the traffic information system and the SSU placement approach with genetic algorithms. In a very similar way, it could be applied to find good locations of supporting units for any other real city, given that a sufficiently detailed road and traffic model has been made available.

6.4.1 Simulation Setup

Likewise as in Section 4.2.1, we utilize VISSIM to simulate vehicular traffic in the city of Brunswick, Germany. We use an average equipment density of 0.25 equipped vehicles per radio range. This corresponds to a VANET equipment ratio of 5 % of all cars. In ns-2, IEEE 802.11 is employed as the MAC protocol, with the two-ray ground propagation model, a communication range of 250 meters, and a carrier sense range of 550 meters. The network simulator also uses the obstacle model that does not allow radio signals to propagate through the walls of buildings as introduced in Section 3.2.

As mentioned above, the objective of the genetic algorithm is to find the optimal vector of supporting units for 100 predefined possible locations (this could, e. g., be the positions where potential cooperation partners are located, who would allow for a SSU to be installed). The genetic algorithm starts with 40 individuals, split into two populations. Each simulation run of the application simulator uses a randomly chosen random seed. This ensures that the process does not get stuck in a local optimum. Our implementation uses a maximum of 100 generations. If this number is reached before homogeneity has occurred, the algorithm is stopped. This did, however, not happen in our simulations.

6.4.2 Travel Time Savings

It seems obvious that placing more supporting units in the scenario improves the performance of the dissemination process and hence higher travel time savings can be achieved. In Figure 6.6, this expectation is confirmed. On the x-axis the number of used supporting units is shown. The y-axis shows the relative travel time for the best supporting unit vector found by the genetic algorithm. The error bars show 99.9 % confidence intervals. A value of one means that no savings can be achieved compared to the current travel time on the optimal standard route, i. e., $\|r^*\| = \|r^*_{\text{std}}\|$. If supporting units are placed at all 100 considered locations, an average car needs a relative travel time of 0.9 compared to the standard travel time. This is equivalent to a travel time saving of 10 %.

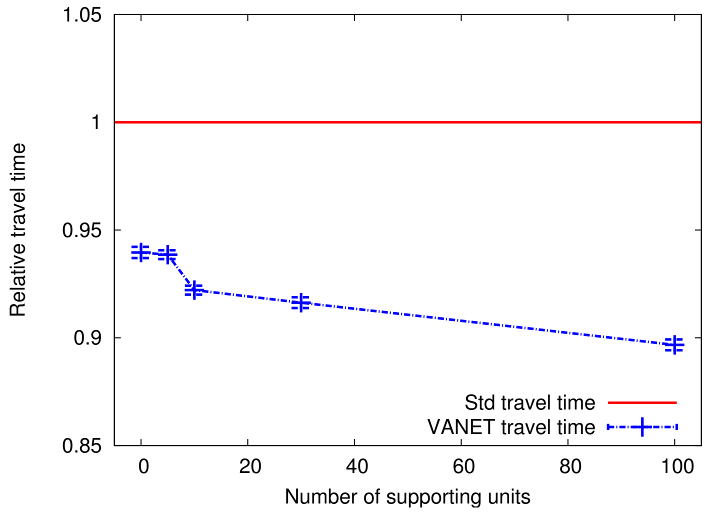


Figure 6.6: Performance evaluation of different active supporting units.

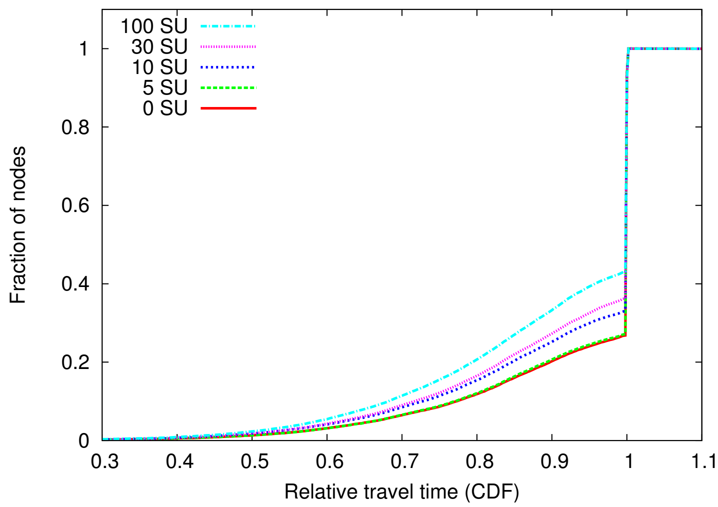


Figure 6.7: Cumulative distribution function of the relative travel times.

Similar time reductions, however, are also possible with fewer supporting units. Even without any infrastructure support the aggregation based dissemination scheme is able to deliver data to cars that can help to improve their routes. The knee in the plot indicates that a good tradeoff between cost and utility in the considered city could be between 10 and 30 SSUs.

It should be noted that a large number of cars does not profit from the additional information, since the standard path to their destination is not congested, or despite a certain level of congestion no better alternative route exists. Those vehicles would not profit from any traffic information system at that time. However, they are included in the calculation of the average travel time savings. Cars for which better routes actually do exist often exhibit substantially larger improvements than the above average values.

This can be seen by investigating the distribution of travel time savings. Figure 6.7 shows the cumulative distribution function of the individual relative travel times. The large fraction of cars with a relative travel time of one includes all those cars that would choose the same path without any dynamic information.

6.4.3 Genetic Algorithm Evolution

Figures 6.8 and 6.9 depict the evolution of the vector of supporting units while the genetic algorithm is running. On the x-axis the IDs of all possible SSU locations are shown. The z-axis, on the right hand side of the figures, shows the progression of generations. For each possible position of a supporting unit in each generation, the z axis shows the number of individuals in which the respective location is occupied by a SSU. Initially, the SSU vectors are chosen randomly, so in the first generations, the supporting units are distributed very homogeneously over the locations. After some generations, however, clear trends become visible and the individuals start to become more and more similar. Some places are virtually completely abandoned quite early. At some point a specific combination of SSU positions becomes predominant and the genetic algorithm cannot gain any further improvement.

A final evaluation of the results of the genetic algorithm is depicted in Figures 6.10 and 6.11. Here, the locations of the supporting units are shown in the analyzed scenario. The crosses represent the 100 possible SSU locations. The locations chosen by the genetic algorithm are marked using squares. It is conspicuous that SSUs are distributed quite uniformly around the city center, in particular if few of them are available. When more supporting units are available, like in Figure 6.11, SSUs are also placed within the city center.

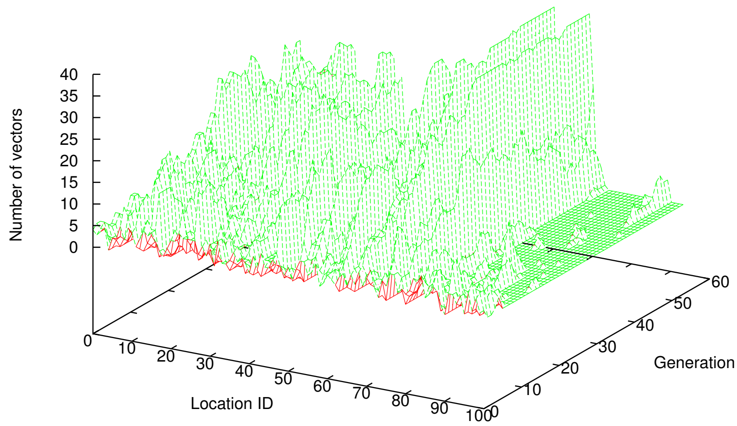


Figure 6.8: Evolution of SSU vectors with ten active supporting units.

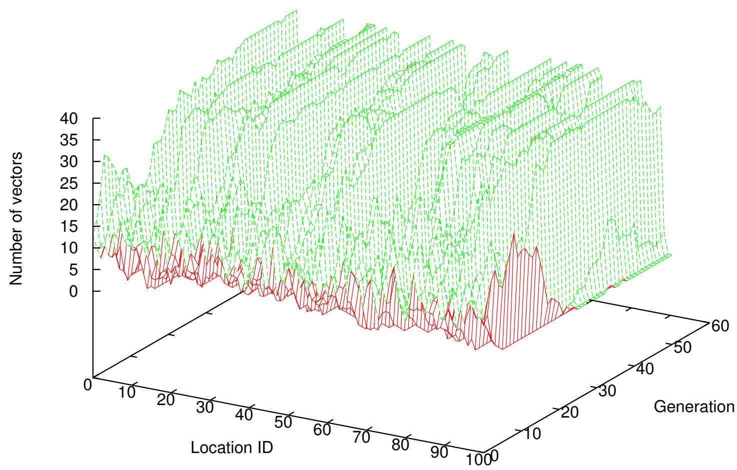


Figure 6.9: Evolution of SSU vectors with thirty active supporting units.

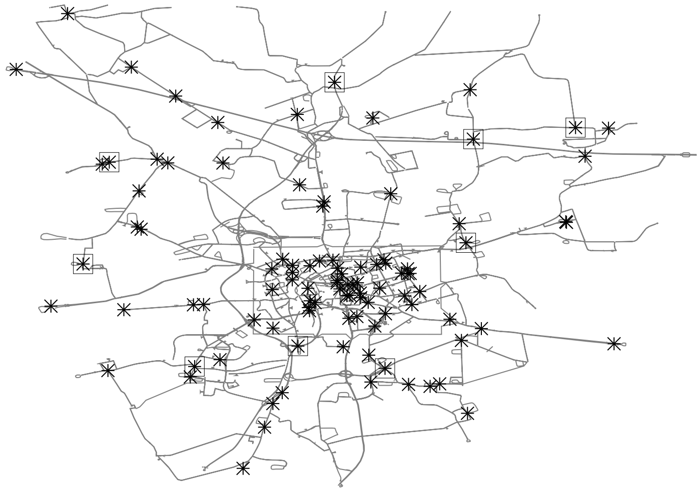


Figure 6.10: Placement of ten active supporting units.

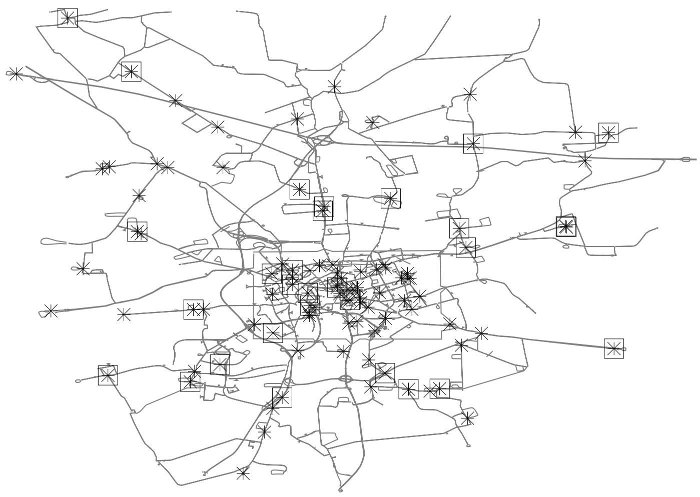


Figure 6.11: Placement of thirty active supporting units.

6.5 Conclusion

In this chapter, we have presented an aggregation scheme for travel time data in road networks. In order to disseminate information within a large network, aggregation is done by means of a multi-layer hierarchy of approximations of the road network. A landmark based aggregation scheme distributes information about the travel times between prominent points of the considered network in order to build an abstract view of more distant regions.

Given this aggregation scheme, it then becomes possible to tackle the second big issue in a VANET-based traffic information system: how and where infrastructure should be used in order to improve information dissemination over larger distances. We have introduced an approach for optimizing the placement of networked roadside infrastructure—(stationary) supporting units—based on genetic algorithms. By a simulation methodology that separates movement and network issues from application behavior it becomes possible to estimate the travel time savings achieved by a given vector of active SSU locations. These savings can be used as a fitness metric, making an application-centric optimization approach feasible. We have confirmed the viability of this approach and assessed the achievable improvements by applying it to a large-scale city VANET model.

Chapter 7

Probabilistic Aggregation for Data Dissemination in VANETs

In order to disseminate traffic information in a comparatively large area, aggregation mechanisms have been proposed to keep the amount of data at a low level. In the last chapter, we have presented an approach for the *hierarchical aggregation* of data that allows to transmit current information on road situations in an efficient way. Nevertheless, other applications in the context of VANETs have been designed. For instance, Caliskan et al. present in [CGM06] a system to exchange information on free parking places [CGM06]. To this group also belong TrafficView [NDLI04b], SOTIS [WER+03b, WER05], CASCADE [IW08a], or StreetSmart [DJ07]. They all have in common that cars autonomously perform measurements about the environment that is then disseminated to other vehicles.

Typically, this is accomplished in the following way. Each car makes observations. An observation is essentially some measured value (traffic density, free parking places,...), related to a position in space (i. e., a road segment or a small area) and a point in time when the observation has been made. All or part of the locally stored information is periodically sent in a beacon based on single-hop broadcasts. Upon reception of such a beacon, a node incorporates the received data into the local knowledge base. By comparing the timestamps of observations, it can ensure that always the most up-to-date value for each position is stored and redistributed. However, if we assume that the spatial density of points for which observations are made is approximately constant, the amount of data increases quadratically with the covered radius. Thus, the amount of data to be broadcasted by each car will likewise increase quickly. This is fatal for the scalability of such a system.

To overcome this problem, the use of hierarchical data aggregation has been proposed: with increasing distance, observations concerning larger and larger areas (or road segment lengths) are combined into one single value. Such an aggregated value could, for example, be the average speed on a longer road segment, or the percentage of free parking places in a part of a city. Coarse aggregates are made available at greater distances, more detailed data is kept only in the near vicinity.

A fundamental issue that arises is that aggregates can not, like single observations, be directly compared with respect to the up-to-dateness and completeness of the contained data. They are created by cars that will typically not have the most up-to-date measurements for all underlying points available. Therefore, multiple aggregates for the same area may exist, based on different, but likely overlapping knowledge. To decide which one is based on "better" underlying data is hard, if not impossible.

In this chapter, we thus propose an algorithm for the *representation and merging* of aggregates that solves this issue. We achieve this by a special data structure: both single observations and aggregates in our scheme do not carry the value of, e. g., the number of free parking places directly, but instead contain an approximation of it in form of a modified Flajolet-Martin sketch [FM85]. This still does not provide a way to compare the quality of two aggregates directly, but it allows for something even better: in our scheme, multiple aggregates for the same area can be merged, yielding a new one that incorporates all the information contained in any one of the aggregates. This is fundamentally different from existing approaches where two aggregates describing the same area cannot be merged[1]. In our scheme it is no longer needed to decide which aggregate contained more up-to-date information since the resulting aggregate comprises all the information from all aggregates that have been merged.

Our approach also allows observations or lower-level aggregates to be integrated into an already existing higher-level aggregate at any time. Note that this is not feasible with existing approaches, because it cannot be determined which data is already present in the aggregate and interesting aggregates like sums or averages are typically duplicate sensitive.

Apart from making decisions regarding the aggregate quality unnecessary, the proposed scheme also largely eases the generation of good aggregates. In order to create a sensible aggregate, a node would usually have to collect data on a significant fraction of the covered area before an aggregate that likely constitutes a good representation can be formed. With our scheme, the aggregate can instead be maintained while being passed around in the network, always incorporating new information on-the-fly. Thus, it may be expected that aggregates in the network are of higher quality.

[1] Any hierarchical aggregation scheme will provide a way to merge lower level aggregates into a higher level aggregate. This is not what we refer to. The point here is that two aggregates describing the same area are merged.

The remainder of this chapter is structured as follows. In Section 7.1, we review some previous uses of Flajolet-Martin sketches in the networking area. Thereafter, we introduce the concept of Flajolet-Martin sketches in Section 7.2 and our algorithm in detail in Section 7.3, and propose two extensions in Section 7.4. In Section 7.5, we present and discuss the results of a simulative evaluation of the algorithm in a VANET city scenario. Finally, we conclude this chapter with a summary in Section 7.6.

7.1 Related Work

Considine et al. present in [CLKB04] a mechanism to collect data in a sensor network. Here data from all nodes should be sent to a common sink. As this type of network is prone to packet losses or node failures the mechanism should deal with these problems. The authors thus propose to use Flajolet-Martin sketches for robust in-network aggregation. Since this data structure is appropriate to build sums of inserted distinct elements in a duplicate insensitive fashion it is not needed to know which "route" a packet is followed to the sink. In contrast to the coordinated collection of information towards a sink in the sensor network we consider here in this thesis the continuously updated distribution of information to all nodes. Additionally, in the above study the problem of removing old information does not occur. However, in an information dissemination approach dealing with frequently changing information this is a major challenge which we will solve here.

In [TKC+04], Tao et al. discuss the use of Flajolet-Martin sketches for spatio-temporal database indexes. The intention is to support queries of the form: "How many objects were in region x over the time interval t?" While these techniques could potentially be used to speed up specific queries in a centralized traffic information system, the problem is rather different from what we consider here, and their approaches are not transferable to a distributed, dissemination based system.

Several other data structures for probabilistic, duplicate insensitive counting have been proposed lately. Examples are LogLog sketches [DF03] and the HyperLogLog sketches [FFGM07] presented by Durand and Flajolet as well as Flajolet et al., respectively. In this thesis, we use Flajolet-Martin sketches because they allow to do a modification that enables the removal of old data from the aggregates which is highly needed for a working traffic information system.

7.2 Flajolet-Martin Sketches

A Flajolet-Martin sketch (also called "FM sketch" or in the following simply "sketch") is a data structure for probabilistic counting of distinct elements that has been introduced in [FM85]. It represents an approximation of a positive integer by a bit field $S = s_1, \ldots, s_w$ of length $w \geq 1$. The bit field is initialized to zero at all positions. To add an element x to the sketch, it is mapped to a position by a hash function h with geometrically distributed positive integer output, where

$$P(h(x) = i) = 2^{-i}. \tag{7.1}$$

The entry $s_{h(x)}$ is then set to one. With probability 2^{-w} we have $h(x) > w$; in this case, no operation is performed. A hash function with the necessary properties can easily be derived from a common hash function with uniformly distributed bit string output by using the position of the first 1-bit in the output string as the hash value.

The central result of [FM85] is that an approximation $C(S)$ of the number of distinct elements added to the sketch can be obtained by locating the end of the initial, uninterrupted sequence of ones, given by

$$Z(S) := \min(\{i \in \mathbb{N}_0 \mid i < w \wedge s_{i+1} = 0\} \cup \{w\}) \tag{7.2}$$

by calculating

$$C(S) := \frac{2^{Z(S)}}{\rho}, \tag{7.3}$$

with $\rho \approx 0.775351$.

The variance of $Z(S)$ is quite significant, and thus the approximation is not very accurate. To overcome this, instead of only one sketch a set of sketches can be used to represent a single value, trading off accuracy against memory. The respective technique is called Probabilistic Counting with Stochastic Averaging (PCSA) in [FM85]. With PCSA, each added element is first mapped to one of the sketches by using an equidistributed hash function, and is then added there. If m sketches are used, denoted by $S_1, \ldots, S_m$, the estimate for the total number of distinct items added is then given by

$$C(S_1, \ldots, S_m) := m \cdot \frac{2^{\frac{\sum_{i=1}^{m} Z(S_i)}{m}}}{\rho}. \tag{7.4}$$

But, as Flajolet and Martin also state in [FM85], this formula is rather inaccurate as long as the number of elements is below approximately $10 \cdot m$. According to [SM07], we thus modify (7.4) in the following way:

$$C(S_1, \ldots, S_m) := m \cdot \frac{2^{\frac{\sum_{i=1}^{m} Z(S_i)}{m}} - 2^{\frac{-\kappa \cdot \sum_{i=1}^{m} Z(S_i)}{m}}}{\rho}, \tag{7.5}$$

with $\kappa \approx 1.75$. This alleviates the initial inaccuracies, while otherwise being asymptotically equivalent to (7.4).

According to [KPS92] PCSA yields a standard error of approximately $1.12/\sqrt{m}$. For many VANET applications, sufficiently good approximations are possible at reasonable sizes.

Sketches can be merged to obtain the total number of distinct elements added to any of them by a simple bit-wise OR. Important here is that, by their construction, repeatedly combining the same sketches or adding already present elements again does not change the results, no matter how often or in which order these operations occur.

7.3 An Aggregation Scheme Based on Flajolet-Martin Sketches

7.3.1 Creating and Merging Sketches

For the purpose of discussion, let us consider a specific application. Assume that we are interested in disseminating the number of free parking places. For now, we do not care about the measured values changing over time. As a first step, we use a sketch (or, with PCSA, a set of sketches) for each road segment. We assume that a car is able to observe the current number of free parking places while passing a road segment, e. g., by collecting data from sensors on the parking places, as proposed in [PGP06]. After passing a road segment with ID r and observing n free parking places on it, a car may add the tuples $(r,1),\ldots,(r,n)$ to the sketch for r, by hashing them and setting the respective bits.

The locally stored sketches for the road segments are periodically broadcasted. Upon reception, received and local sketches are merged by calculating the bit-wise OR. Figure 7.1 exemplifies this procedure. Two cars, A and B, make independent observations on the same road segment (with ID 17). A observes four free parking places and thus hashes the tuples $(17,1),\ldots,(17,4)$ into its sketch for road segment 17. B observes five free parking places, and consequently adds $(17,1),\ldots,(17,5)$. If A and B meet later, and A receives a

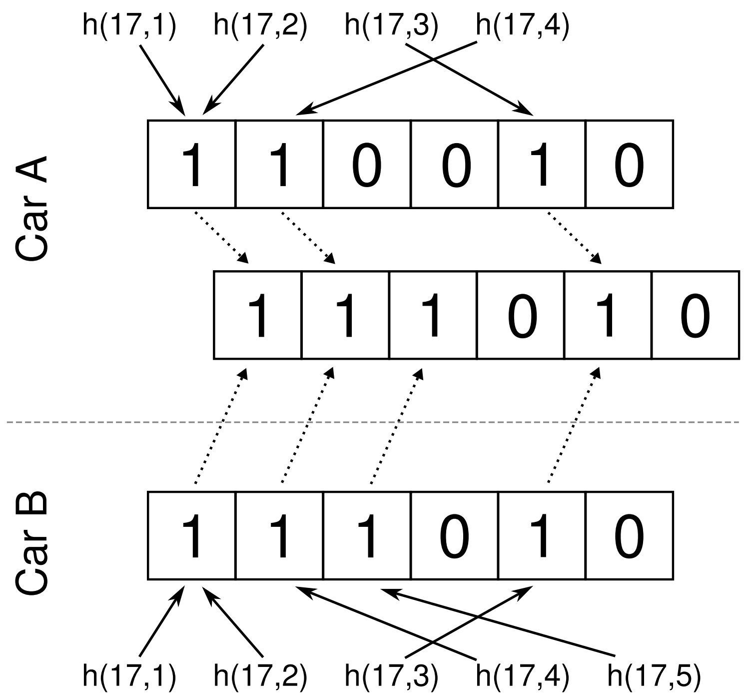


Figure 7.1: Generation and merging of FM sketch aggregates.

transmission containing *B*'s sketch, *A* merges them by bit-wise OR and obtains a new sketch, replacing its previous one.

The hashed tuples (r, i) are identical for different observers, the observed value determines only how many of them are added. If all observers use the same hash function (something that could easily be standardized), the same number of free parking places on the same road segment will set the same bits, a lower number a subset thereof. Of course, in the current basic algorithm, bits that have once been set will never get unset again, and the sketch is therefore not able to follow decreasing values. We will soon see how to extend the data structure in order to overcome this limitation.

7.3.2 Hierarchical Aggregation

Hierarchical aggregation is typically done on trees, often on symmetric and self-similar ones like quad-trees over the two-dimensional plane. But while it may for instance be expected that the traffic situation does not differ much among a set of similar and close-by road segments, or that the fraction of free parking places is relatively constant within a neighborhood, it might at the same time be vastly different not too far away: for example on the other side of a highway or a river. Therefore, a good aggregation scheme should respect the environment it is imposed on. We envision that such an aggregation hierarchy is pre-defined in the map data, following the underlying structure and grouping areas in a way that reflects their natural relations, like city districts or road hierarchies.

This is explicitly supported by our algorithm. Let L denote the set of locations for which observations can be made, like the entirety of road segments, or simply all points on the map. Many aggregates are possible—in principle, any arbitrary combination of locations could be aggregated. Possible aggregates are thus the (non-empty) elements of L's power set $\mathcal{P}(L)$. We may choose a subset $\mathcal{A}$ of these as the areas for which sketches are to be maintained:

$$\mathcal{A} \subseteq \mathcal{P}(L) \setminus \{\emptyset\} \tag{7.6}$$

The structure of $\mathcal{A}$ is not constrained by our method, though certain choices exhibit benefits.

Based on the outlined idea, hierarchical aggregation can be accomplished in the following way. We allot sketches for all elements in $\mathcal{A}$. Any observation made for some location $l \in L$ can immediately be incorporated into each aggregate for which the aggregated area A contains l, that is, $l \in A$. Consequently, in our example application of counting free parking places, these aggregates will contain the total counts in their respective areas.

It should be noted that not necessarily for each location a separate sketch needs to be maintained, i. e., there is no necessity that $\forall l \in L : \{l\} \in \mathcal{A}$. Especially if L is large (or continuous!), it might make sense to maintain sketches only for areas encompassing multiple locations.

Information on small-scale areas will then typically be kept in the closer vicinity and is broadcasted only there, while further away cars will preferably maintain and distribute larger-scale aggregates of the region. The duplicate insensitivity of the sketches allows for aggregates to be merged in just the same way as it has been introduced above for sketches of single locations. But in particular, any received sketch for some area A can immediately be incorporated into any *superordinate* aggregate A', where superordinate means that A is wholly covered by A', i. e., $A \subseteq A'$.

To allow for incorporating received information into as many aggregates as possible, $\mathcal{A}$ will thus indeed often be a hierarchical tree structure, where for all $A_1, A_2 \in \mathcal{A}$ it holds that

$$A_1 \cap A_2 \neq \emptyset \quad \Longrightarrow \quad A_1 \subseteq A_2 \ \vee \ A_2 \subseteq A_1. \tag{7.7}$$

Even this, however, implies neither any symmetry nor the same depth of all subtrees of any given node. It also does not exclude cases where some higher-level area is not

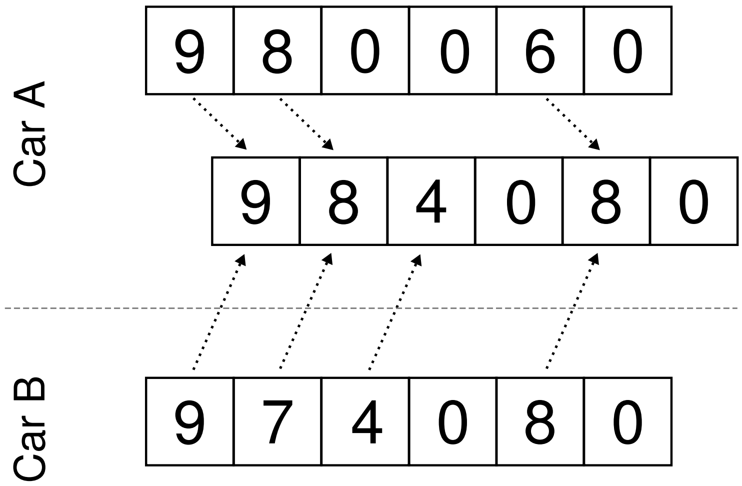


Figure 7.2: Merging of soft-state sketches.

completely covered by smaller subareas. Therefore, this concept is much more powerful than the commonly used aggregation trees.

7.3.3 Soft-state Sketches

With the so far discussed algorithm, the sketches will always represent the maximum of all ever observed values for each road segment. This is of course not desirable. Therefore, a method is needed to remove old observations.

We accomplish that by modifying the original FM sketches. We use small counters of n bits length instead of single bits at each index position. These counters represent a time to live (TTL) in the range $0,\dots,2^n-1$ for that bit. The operation of setting a bit to one after an observation is replaced by setting the corresponding counter to the maximum TTL, to $T := 2^n - 1$. Broadcasts containing the sketches are sent at regular intervals. Just before sending such a broadcast with information from the local knowledge base, all counters in the locally maintained sketches are decremented by one, if they are not yet zero.

When incorporating a received sketch into the local knowledge base, the bit-wise OR is substituted by a position-wise maximum operation. This yields a soft-state variant of FM sketches, in which previously inserted elements essentially die out after their TTL has expired, unless they are refreshed by a newer observation. The merging is visualized in Figure 7.2. Car A receives an aggregate from car B, and updates its own soft-state sketch accordingly.

For obtaining the current value from a soft-state sketch, the algorithm remains essentially unchanged; still, the smallest index position with value zero is identified and used. Note that this incurs some delay if a bit position is no longer set in newer observations. Coming back to our above example of observing parking places, assume that no further observations are made which set a particular bit position (e. g., because there is no longer a free parking place being hashed to it). If the position had previously been set in an aggregate, then the TTL value will decrease over time until it arrives at zero.

As an extension, it is possible to consider the smallest index position with a value below some threshold instead of the smallest position being zero, by using

$$Z_t(S) := \min(\{i \in \mathbb{N}_0 \mid i < w \wedge s_{i+1} \leq T - t\} \cup \{w\}) \tag{7.8}$$

in the role of $Z(S)$—which equals $Z_T(S)$—in (7.3) or (7.5). The threshold t can be chosen arbitrarily and individually whenever evaluating a sketch, in the range between 1 and T. This yields an approximation $C_t(S)$ of the total distinct item count observed in the last t broadcasting intervals. It therefore allows to dynamically choose a "cutoff horizon", thereby trading off between taking only more recent observations—that is, the most up-to-date data—into account and working on a larger data basis.

7.3.4 Example Applications and Practical Issues

The simple parking place counting application as discussed above has a major drawback. In case of an aggregate showing a small total number of free parking places, it is not clear whether this is caused by a small number of parking places being free, or by a small number of free parking places having been *observed* in the time interval covered by the TTL, due to a generally low number of observations. Note that this is not a problem of our approach, but a general one. It is, fortunately, relatively easy to overcome. In the proposed application a car may distribute *two* values with separate sketches instead of just one: the number of observed free parking places, and in addition the number of observed total parking places. Both values in combination describe which *fraction* of the observed parking places is free. It generally seems that distributing such relative values is more robust. Recall that due to the soft-state approach it is not necessary to report occupied parking places since a no longer free parking place will die out if it is not refreshed.

This also accommodates trading off the considered timespan as introduced above: the application can easily infer the comprehensiveness and coverage of the underlying data basis for increasing time horizons, to optimize the tradeoff.

Sketches can be used to approximate sums of positive integers, but can be generalized to general integers and fixed or floating point numbers [CLKB04]. Our scheme is thus applicable whenever the aggregated value can be expressed through sums. Examples are counts, sums, or averages, but also variance and standard deviation (through the average and the average of the squares) or even products (by adding logarithms) [CLKB04]. The accuracies of the approximations, of course, vary, and an appropriate tradeoff for the specific application has to be found.

Further application examples could be the dissemination of the current traffic density (e. g., by distributing the number of observed vehicles and the total length of the roads for which there are observations), or the current average speed on a road. Both are useful to support navigation and route planning.

7.4 Extensions

7.4.1 Compressing Soft-state Sketches

The techniques discussed above allow to summarize individual observations into hierarchical aggregates of increasing geographical scope. This reduces the number of data items that need to be exchanged in the network. However, the size of the individual soft-state sketches themselves is relatively large: each individual aggregate consists of $m \cdot w$ TTL counters. So, sketches in their standard representation consume significant network bandwidth when they are transmitted. This raises the question whether the size of transmitted sketches and thereby the bandwidth requirements can be reduced.

A close look easily reveals that sketches indeed carry a lot of redundancy. Positions on the left hand side of the sketch are much more likely to be “hit” when an item is added. In soft-state sketches, these entries are thus likely to have a high remaining TTL, i. e., a high value. On the right hand side of the sketch, in contrast, additions occur comparatively seldom, and it may be expected that many entries are zero. This redundant structure indicates that effective data compression is possible.

The local storage size of the knowledge base is much less critical than the network capacity constraints. Thus, soft-state sketches need not be compressed while being stored locally, facilitating an easy implementation of the update, merge and decay operations discussed above. We envision that a compression mechanism is used for the encoding and decoding of sketches when they are transmitted in beacons. A well-suited compression mechanism will therefore allow for efficient on-the-fly compression and decompression of sketches upon sending or receiving beacons, without causing too much computational overhead.

In [SM07], a compression scheme for standard FM sketches with these characteristics has been proposed. The idea is to transmit the total number of leading ones in a PCSA set first. This value characterizes the probability distribution of the possible values (0 and 1) at each bit position. Both sender and receiver use this as a data model for arithmetic coding [RL79], thereby achieving compression very close to the entropy limit in an algorithmically very efficient way. Unfortunately, this idea cannot directly be applied to soft-state sketches, because individual entries in soft-state sketches can take more than two values, and the probability distributions cannot be parameterized as compactly as in the case of standard FM sketches.

Nevertheless, we may take up the general idea of modelling the interdependencies of the sketch entries' probability distributions, thereby exploiting our knowledge about the structure of the soft-state sketches for efficient compression. The task of a data model for arithmetic coding is to make "predictions" about the probabilities of all possible values for the next input character, given the input so far. The model is used by both encoder and decoder, so that they agree on the same probabilities. For good results, the quality of the model is crucial: better prediction results in better compression. So, given that we are able to formulate a model that makes good predictions about the values of the individual soft-state sketch entries, arithmetic coding constitutes an ideal basis for a tailored compression scheme.

In order to come up with such a model, we examine the structure of a PCSA set of soft-state sketches with maximum TTL T more closely. For all $t, 1 \le t \le T$, let x_t be the (unknown) number of distinct elements that have been added to the PCSA set, for which the remaining TTL is t (i. e., x_t distinct elements that have been added $T - t$ intervals ago). Let

$$X_t := \sum_{i=t}^{T} x_i. \tag{7.9}$$

In a PCSA set with m component sketches, the probability that the j-th entry in the i-th sketch, here denoted by $s_{i,j}$, is "hit" by an insertion is $2^{-j}/m$. Hence, the probability that, after X_t insertions of elements with TTL $\ge t$ position $s_{i,j}$ still has a TTL below t is

$$p_{j,t} := P(s_{i,j} < t) = \left(1 - \frac{2^{-j}}{m}\right)^{X_t}. \tag{7.10}$$

Note that $p_{j,t}$ does not depend on i, which is clear from the fact that there is no structural difference between the individual sketches in a PCSA set. I. e., the probability of a certain value does not depend on the index of the sketch in the PCSA set, but only on the position within the sketch.

For the entry one step to the left from $s_{i,j}$, i. e., for $s_{i,j-1}$, the respective probability is

$$p_{j-1,t} = \left(1 - 2 \cdot \frac{2^{-j}}{m}\right)^{X_t}. \tag{7.11}$$

By solving (7.10) for X_t and then using the result in (7.11) we obtain

$$\begin{aligned} p_{j-1,t} &= \left(1 - 2 \cdot \frac{2^{-j}}{m}\right)^{\frac{\ln(p_{j,t})}{\ln(1-(2^{-j}/m))}} \\ &= \left(p_{j,t}\right)^{\frac{\ln(1-2\cdot(2^{-j}/m))}{\ln(1-(2^{-j}/m))}}. \end{aligned} \tag{7.12}$$

Thus, based on the probability that the sketch entry $s_{i,j}$ has a value of less than t, we can now calculate the probability that entry $s_{i,j-1}$ is less than t. Therefore, we know how the probability distributions for the sketch entries in neighboring positions are interrelated. However, this does of course *not* tell us what the actual probabilities $p_{j,t}$ are. They depend on the individual sketch being processed.

Witten et al. [WNC87] use arithmetic coding with a dynamic model. In their scheme, the input distribution is "learned" while processing the input. Basically, the algorithm keeps track of the input distribution in the data processed so far, continuously adjusting the model. However, this approach assumes that the input distribution is the same for all input characters—an assumption that does not hold when compressing soft-state sketches.

In our sketch compression scheme, we combine the idea of learning the input distribution on-the-fly with our knowledge about the interdependencies between the distributions at different positions in the sketch. We may think of a PCSA set as a matrix, as in Figure 7.3, where an example consisting of three sketches with six entries each is shown. In our algorithm, we traverse the PCSA set "column-wise" from right to left, i. e., we start with the rightmost entries of all component sketches. Because these entries all exhibit the same probability distribution ($p_{j,t}$ does not depend on i), we may apply Witten at al.'s algorithm within this part of the matrix. While processing the entries, the algorithm counts the number of occurrences of each value and builds a continuously refined model for this column.

However, this model does not directly apply to other columns. Before proceeding with the next column, we make use of the interrelation of the probability distributions expressed in (7.12). Based on the probability distribution estimated by Witten and Neil's algorithm for column j, we can calculate an estimate for the probability distribution in column $j-1$. We then in turn use this transformed distribution as a starting point for the compression of column $j-1$. While the entries in this column are compressed, the

S_1	9	8	4	2	0	0
S_2	9	6	5	0	1	0
S_3	9	9	7	1	0	0

Figure 7.3: A PCSA set consisting of three soft-state sketches.

model is again continuously refined and adjusted to the actual values, before it is again transformed according to (7.12) when transitioning to column $j-2$, and so on.

However, while (7.12) is an exact representation of the interdependencies between the probability distributions, it is not very well-suited for practical purposes, because its evaluation requires complex floating-point operations. We observe that for $m \to \infty$, (7.12) converges according to the rule of BERNOULLI-L'HOSPITAL to

$$\lim_{m\to\infty} p_{j-1,t} = p_{j,t}^2. \tag{7.13}$$

Because the convergence is very quick, one may employ the much simpler formula

$$p_{j-1,t} = p_{j,t}^2 \tag{7.14}$$

in practical implementations, as a very good approximation. If (7.14) is used for transforming the estimated probabilities upon column transitions in combination with Witten et al.'s arithmetic coding algorithm, all steps necessary for compression and decompression can be performed with pure integer arithmetics in linear time. It is therefore ideally suited even for resource-constrained devices.

7.4.2 Longer Counters for Larger Aggregates

In a typical application of our aggregation scheme, large aggregates will be distributed over longer distances, while smaller aggregates remain in the closer vicinity. But distribution over long distances implies that the aggregates are longer underway, and will typically traverse more hops. In short: the information will be older when it arrives at the place where it is used. Consequently, while locally relatively quick ageing of the information will be tolerable (or even desirable), for larger, widely distributed aggregates it will likely be advisable to use a longer lifetime for the soft-state information. To accomplish this, it is possible to extend our algorithm in a way that uses soft-state sketches with longer TTL counters for larger aggregates. Obviously, this increases the size of these aggregates, but since the available TTL range grows exponentially with the counter size it scales very well.

Using different counter lengths in different aggregates slightly increases the complexity of the merge operation. However, the necessary modifications become relatively straightforward if the *age* of an entry is considered instead of its TTL. The age is the number of decrements that have occurred at that position, i. e., its difference from the maximum TTL.

Instead of setting the position in the merged aggregate to the maximum TTL of the two merged soft-state sketches, it is set to the TTL corresponding to the *minimum age*, where the received TTL is larger than zero. The resulting operation is equivalent in the case of identical counter sizes in the two sketches, but differs if the counter sizes are different.

As an example, consider a locally stored aggregate with a counter size of eight bits. Let us focus on one single position, and assume that it currently has a value of 8. This corresponds to an age of $255-8=247$, since the maximum TTL is $2^8-1=255$ here. Now a sketch for a sub-area contained in our aggregate is received, which uses a counter size of only four bits. The value at our bit position is 10 in this sketch. Because the maximum TTL is 15, this corresponds to an age of 5. Since the minimum age of local and received aggregate is 5, we set the local aggregate to this age, and consequently to a TTL value of $255-5=250$.

More formally speaking, if T_{local} and s_{local} are the maximum TTL and the current value of an entry of the locally stored sketch, and T_{recv} and $s_{\text{recv}}>0$ are the corresponding counterparts in a received sketch that is to be merged in, then the new value of the respective position is given by

$$\begin{aligned} T_{\text{local}} - \min\{T_{\text{local}} - s_{\text{local}},\ T_{\text{recv}} - s_{\text{recv}}\} \\ = \max\{s_{\text{local}},\ s_{\text{recv}} + T_{\text{local}} - T_{\text{recv}}\}. \end{aligned} \tag{7.15}$$

This scheme is, of course, applicable for arbitrary combinations of different counter sizes.

7.5 Evaluation

7.5.1 Methodology

To evaluate our scheme, we implemented it in the simulation environment encompassing the network simulator ns-2 [ns2] and the microscopic traffic simulator VISSIM [PTV] as presented in Chapter 3 in association with the Brunswick traffic model introduced in Section 4.2.1.

We simulate a VANET equipment penetration ratio of 20 %. This corresponds to an average equipment density of one vehicle per communication range. In ns-2, IEEE 802.11 is employed as the MAC protocol, with the two-ray ground propagation model with a communication range of 250 meters and a carrier sense range of 550 meters. Again, the network simulator is enhanced with the obstacle modeling that does not allow radio signals to propagate through the walls of buildings mentioned in Section 3.2.

Because, in a model of the given size, a combined simulation of all aspects in parallel is extremely time-consuming and costly in terms of computation, we break it down to a three-step process. First, vehicle movements are generated using VISSIM. Subsequently, the beaconing process is simulated by ns-2. In this step, all VANET-equipped cars periodically send beacons of size 1096 byte (1024 byte payload plus headers) once every five seconds. The PHY and MAC models in ns-2 decide which of these beacons are received by which subset of the cars—note that this is not affected by the actual data contained in the beacons. Finally, the application logic is simulated, including the information exchanged in the beacons and the knowledge base of each car before and after each beacon transmission.

The road density, movement speed and pattern, etc. are, like in a real city, very heterogeneous in the simulation model. This makes the model realistic. But evaluating a protocol in such a complex environment in absolute terms is difficult. For example, if a high-speed road with lots of traffic connects two points which are 5 km apart, then one can expect well-working dissemination over that distance. If two other points with the same distance are separated by a municipal park, then the performance is bound to be worse. Often, it can make a large difference whether a single vehicle carrying a piece of information gets through to a certain point and distributes its knowledge there.

In order to overcome these difficulties at least partially, we opted to use an optimal reference protocol in an otherwise exactly identical simulation setting for comparison. In this protocol, information is spread by periodic beacons, in just the same way as in our and many other approaches. But the optimal protocol does not care about practical bandwidth limitations.

In a simulator, the packet size and the amount of information actually exchanged between the nodes are independent. The reference protocol exploits this fact and does *not* aggregate information at all. Instead, a received "aggregate" contains the sending node's most up-to-date measurement values of all the locations. Each of these values also carries an individual timestamp, making optimal merging of information trivial. In order to avoid keeping too old measurements in the network, we use a timeout equivalent to the initial TTL of the soft-state sketches' entries.

Obviously, implementing this optimal reference protocol is easily possible in a simulator, but not in practice. However, it is well suited as a benchmark: with a practical protocol based on beacons and a knowledge base the cars can never have got better information.

We evaluate both our scheme and the optimal reference protocol with an idealized application. We subdivide the city area into 256 small areas, which we use as single locations. We simulate a simple stochastic process for each of these areas, the current value of which can be "measured" by every car entering the respective area. This could be interpreted as "counting" the number of free parking places in this area.

Our simulations cover a timespan of 15 minutes. Aggregation generally makes sense only if the dynamics of areas which are geographically close together are not completely unrelated (which can of course be expected in the real world). Thus, the stochastic processes of the single locations are also not totally independent. All of them typically start relatively low, and increase substantially over the first half of the simulation, before they subsequently tend to decrease again.

In relation to the relatively short total time, the changes are very rapid, and thus are very challenging for a dissemination protocol that needs to keep track of them. Since, as discussed above, disseminating relative values is much more appropriate than absolute sums, we also have one reference value per area—e. g., the "total number of parking places"—, which is likewise "observed" and distributed.

For hierarchical aggregation, groups of four neighboring areas are combined into a total of 64 medium-sized areas, and these are in the same way aggregated further to form 16 large-scale areas. For deciding which information is to be transmitted in a beacon, we constrain ourselves to a simple selection strategy—mainly to limit the overall complexity of the system and to allow reasonable result interpretation while regarding the results of Chapter 5. In our simulations, each node transmits at most 27 aggregates in each beacon: on all three hierarchy levels, the area the car is currently in and the eight neighboring areas are sent.

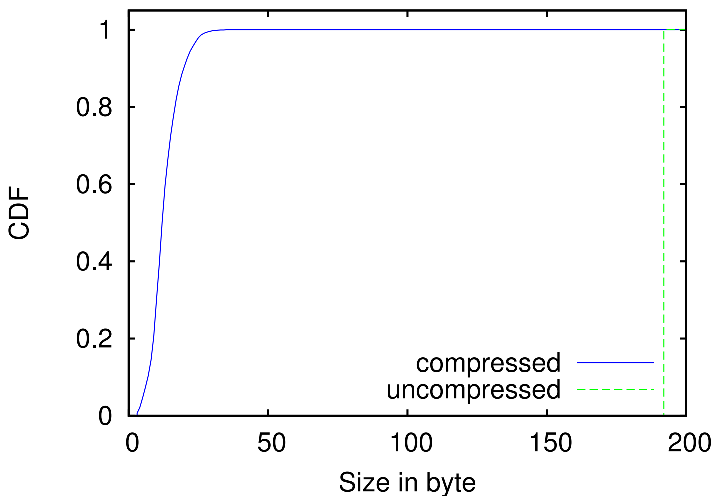


Figure 7.4: CDF of compressed sketch sizes.

7.5.2 Sketch Size

Before we look at the information *within* the aggregates, let us first have a quick glance at the performance of the compression scheme discussed in Section 7.4.1. In Figure 7.4, it is depicted the cumulative distribution function (CDF) of the compressed sizes of the aggregates which are transmitted in our simulation scenario.

The dashed line shows the uncompressed size of the sketches, which is, of course, constant. We use a PCSA set of $m = 16$ soft-state sketches with length $w = 24$ and $n = 5$ bits per counter, i. e., the maximum TTL is $T = 31$. Therefore, the uncompressed size of an aggregate is 240 byte. From the figure, it becomes obvious that the compression yields a very significant size reduction: most sketches are compressed down to a size of 10–50 bytes with an average size of 19 byte.

7.5.3 Local Accuracy

We now evaluate how well the aggregates reproduce the current value of a measured parameter *locally*, i. e., at the location where the measurement is performed. This is of interest because the data representation with soft-state sketches is probabilistic, and therefore does not necessarily reproduce the current value exactly.

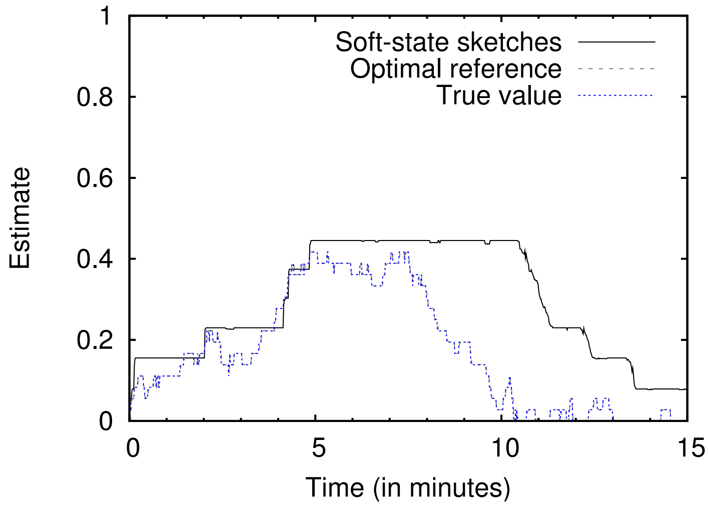


Figure 7.5: Accuracy of local measurement representation.

Figure 7.5 shows, for one typical location, the average value from the knowledge bases of all vehicles that are currently within the corresponding region. As discussed above, we use a relative value, given by the ratio of the number currently free “parking places”, which is changing over time, and their (fixed) total number for this location. For comparison, we also plot the current true value, and we perform the same evaluation with the optimal dissemination protocol. In the specific case of a single, locally continuously measurable value the latter two are of course always identical.

The figure shows that the sketch-based dissemination is able to model the correct value quite well. When the measured value starts decreasing again, the time needed for the soft-state decaying of the no longer set bits becomes visible: the sketch represents the maximum value observed in the recent past. Thus, there is a time lag before it follows a decreasing value. Due to the beaconing frequency of one beacon every five seconds and the maximum TTL of 31 beaconing cycles it is about $31 \cdot 5\,\text{s} \approx 2.5\,\text{min}$. Recall in this context that an application can dynamically select the cutoff horizon in (7.8), and may therefore tune this parameter locally and individually at any time.

Taking the time lag into account, the soft-state sketches indeed reflect the true situation very well here. This encouraging first result leads us to the next question: how well can aggregates for larger regions be formed and maintained?

7.5.4 Forming Aggregates

With the proposed scheme, aggregates for larger regions can and will be formed wherever information on the respective region flows together. But nevertheless it is reasonable to expect that this will most regularly happen within the respective region. Therefore, we now look at how well sketch-based aggregates represent the situation in a larger area, while they are stored, passed around, and merged by cars within this area. Note that now the cars can no longer observe the entirety of the underlying information themselves. They can only measure the current value of their own location, and thus depend on received and merged information from other nodes in order to complete their picture.

In Figures 7.6 and 7.7, we show such evaluations for a typical medium-sized and a large inner-city aggregate (consisting of 4 and 16 locations, respectively). It is visible that perfect knowledge like in Figure 7.5 is no longer possible: the optimal reference protocol does not always have perfect information. Especially for the high-level aggregate in Figure 7.7, the inevitable delay until up-to-date information on the entirety of the area has arrived can clearly be seen from the offset between the true value and the optimal protocol. The sketch-based aggregation is close to the optimal knowledge, again with the previously discussed soft-state time lag in case of a decreasing value. These results demonstrate that probabilistic aggregation is indeed able to collect the available information, yielding aggregates that represent the available knowledge.

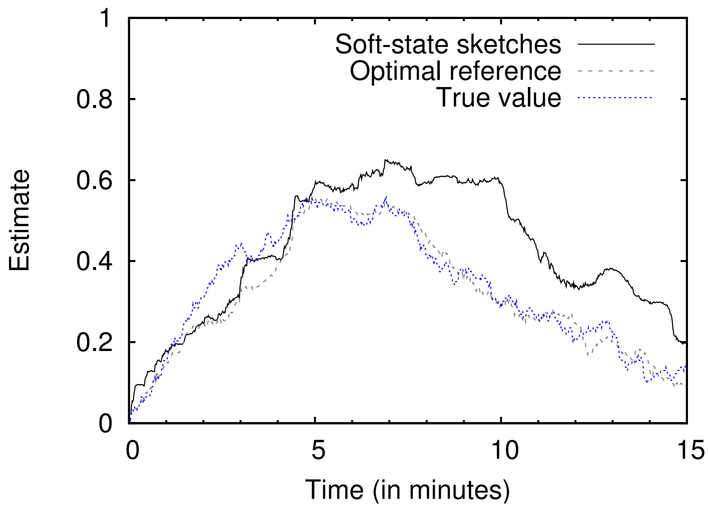


Figure 7.6: Accuracy of local medium-sized aggregate.

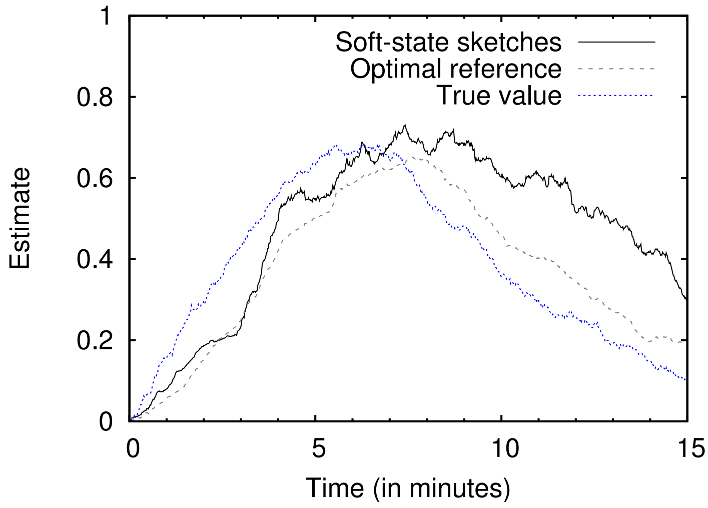


Figure 7.7: Accuracy of local large aggregate.

7.5.5 Distributing Aggregates

Now, we look at further away regions and evaluate the representation of the situation in the knowledge bases of participating cars. Figures 7.8 and 7.9 represent two instances of medium-sized regions, as they are seen at a distance of about 3 km beeline. In addition to the time lag for the information transport it is not astonishing that even the optimal reference protocol does not always have complete information. If up-to-date data only on parts of the total aggregation area is present and this data is not typical for the whole aggregate, effects like the overestimation around simulation minute six in Figure 7.9 are the logical consequence. Nevertheless, the estimates once again reflect the true situation in the modeled regions well. Finally, this is also confirmed by a look at a large aggregate's representation of the outskirt area in our model, as it is depicted in Figure 7.10.

In summary, sketch-based probabilistic aggregation can be used to create aggregates that come close to what can theoretically be achieved with the considered kind of system. This optimum was here represented by a non-realizable optimal reference protocol. We consider these results very encouraging indications that the proposed algorithm is a suitable way to overcome the often observed general difficulties of distributed, uncoordinated data aggregation in dissemination schemes.

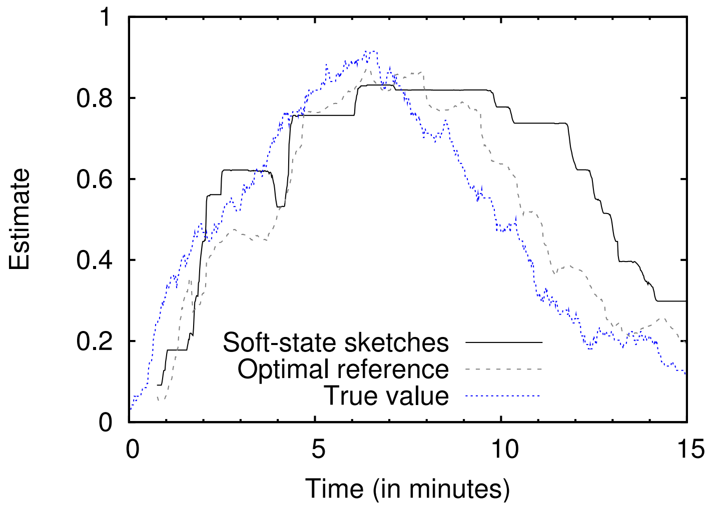


Figure 7.8: Accuracy of distant medium-sized aggregate.

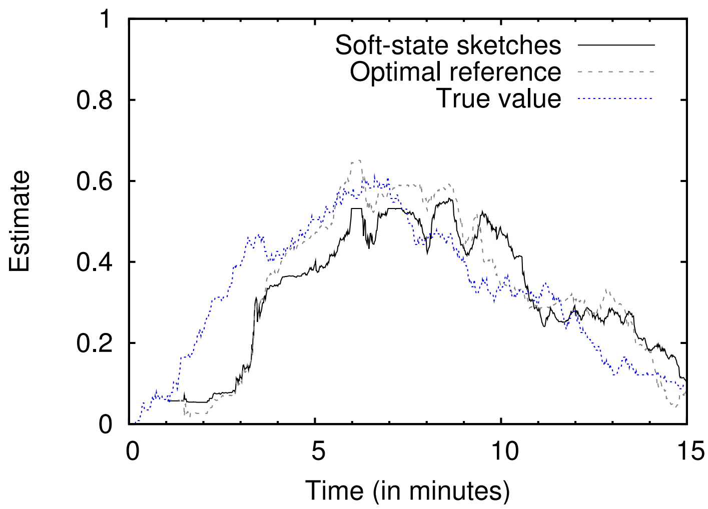


Figure 7.9: Accuracy of distant medium-sized aggregate.

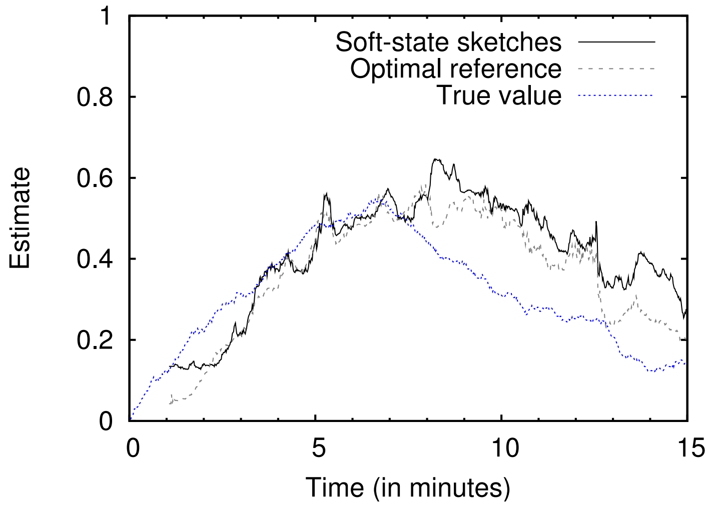


Figure 7.10: Accuracy of distant large aggregate.

7.6 Conclusion

In this chapter, we introduced a data aggregation mechanism for dissemination-based VANET applications. It is based on a probabilistic data representation, Flajolet-Martin sketches, which we extended to a soft-state data structure. This yields duplicate insensitive aggregates, thereby overcoming major unsolved problems in existing aggregation schemes for VANETs. We discussed several application scenarios of our scheme and possible extensions, showing the broad applicability of the approach.

In a simulative evaluation study using ns-2 and a realistically modeled inner-city VANET we assessed the performance of the aggregation scheme. We compared the values to what could ideally be achieved in the same setting. The results of this study confirm our expectation that sketch-based schemes are well-suited for the considered purpose, and may well form a central building block for a large variety of future VANET applications.

Chapter 8

Conclusion

In this thesis we have analyzed information dissemination in vehicular (ad-hoc) networks. In contrast to MANETs that often build upon unicast communication, vehicular networks can be characterized by their many-to-many communication paradigm. We have identified and tackled the two main challenges of this type of networks: dealing with *limited connectivity* and dealing with *limited bandwidth*.

In order to analyze this distinct type of network we first have implemented an integrated meta simulation toolkit which makes it possible to conduct holistic evaluations of moving cars and radio propagation. The toolkit consists of the traffic simulator VISSIM in association with realistic traffic scenarios (Brunswick city scenario) and the network simulator ns-2. With this coupling at hand we were able to examine the effects of mobility on the dissemination of data in a vehicular network. Furthermore, it is now feasible to influence the movement of single cars within the traffic simulator initiated by, e. g., the reception of an information message. This toolkit presented in Chapter 3 has been used in all simulation evaluations of the approaches presented in this thesis.

We were then able to concentrate on the first challenge of information dissemination in vehicular networks. In Chapter 4 we have shown that information can be spread by the locomotion of vehicles carrying this data and by the radio transmission to vehicles in the vicinity. However, due to the relatively low initial equipment density of VANET technology we realized that information dissemination would not be possible in the early phase of rollout. Since applications based on a widespread information dissemination would not get enough data to work properly nobody would buy this technology which would hamper the rollout process. We therefore proposed to use additional infrastructure (supporting units) that allow to span distant regions and can provide passing by vehicles with current information. We have further elaborated that it is necessary that these supporting units are connected to a backbone network and share a common knowledge base.

In Chapter 5 we turned our attention to the second major challenge of information spreading—the limited bandwidth. We have constituted a formal model of the dissemination process. It was concluded that the capacity would not be adequate if almost all vehicles are the processor of, e. g., traffic situation updates which need to be spread atomically to all regions of the network. These insights motivated the usage of aggregation mechanisms to subsume data with respect to the covered distance.

So as to tackle these two challenges we have developed an aggregation scheme in Chapter 6. The main idea of this scheme was to use a multi layered approach. We combined a hierarchical aggregation technique with a *landmark* summarizing technique. The procedure is based on the insight that when building a traffic information system current and detailed information has to be made available in the vicinity of the requesting vehicle. However, information on farther away regions can be much coarser. In addition to the aggregation scheme we have analyzed how to deal with the limited connectivity as well. In order to use a minimum number of supporting units while gaining as high information dissemination speed as possible we have proposed a genetic algorithm for solving this hard optimization problem. The large number of possible combinations where to place some supporting units did not allow to analyze all eligible combinations of placements. However, with the help of the genetic algorithm we were able to find good locations in a short evaluation process. By implementing a prototype navigation system we have shown that with only few supporting units a high benefit for the drivers that would otherwise have encountered a congested route can be achieved.

The last contribution of this thesis is on the representation of disseminated aggregates. In Chapter 7, we have proposed a probabilistic aggregation scheme. The main focus of this work was on the merging of different aggregates. In particular when dealing with overlapping aggregates it is in general not possible to extract information out of one aggregate to insert it into another aggregate. Since aggregation means loosing details on what has been inserted an extraction of data is not possible. We therefore proposed to use soft-state sketches for the counting of observations, e. g., of free parking lots. The main advantage of sketches is in their property to be duplicate insensitive. No matter how often one item is inserted into a sketch the result will always be the same. We have introduced a soft-state variant to erase already inserted elements. By using a die-out process, observations are not deleted explicitly but vanish implicitly. By adopting a compression scheme based on arithmetic coding we have shown that this representation scheme for aggregates can be used in a vehicular information dissemination environment.

In summary, we have analyzed information dissemination in vehicular networks. We divided the evaluation target into several parts. In order to classify these parts we proposed two main challenges that we had to deal with: limited connectivity and limited capacity. We have presented different mechanisms on the aggregation of data and on the performance increase of the connectivity, especially during rollout of VANET technology. With these techniques at hand spreading information on the current traffic situation in a vehicular network becomes possible and allows drivers to adapt their routes accordingly—avoiding the gridlock.

Bibliography

Own Publications

[BJL+08] Jens Bendisposto, Michael Jastram, Michael Leuschel, Christian Lochert, Björn Scheuermann, and Ingo Weigelt. Validating wireless congestion control and reliability protocols using ProB and Rodin. In *FMWS '08: Proceedings of the International Workshop on Formal Methods for Wireless Systems*, pages 77–81, August 2008.

[FMH+05] Holger Füßler, Martin Mauve, Hannes Hartenstein, Christian Lochert, Dieter Vollmer, Dagmar Herrmann, and Walter Franz. Position-Based Routing in Ad-Hoc Wireless Networks. In Walter Franz, Hannes Hartenstein, and Martin Mauve, editors, *Inter-Vehicle-Communications Based on Ad Hoc Networking Principles —The FleetNet Project*, pages 117–143. Universitätsverlag Karlsruhe, Karlsruhe, Germany, November 2005.

[JLSM08] Yves Igor Jerschow, Christian Lochert, Björn Scheuermann, and Martin Mauve. CLL: A cryptographic link layer for local area networks. In *SCN '08: Proceedings of the 6th Conference on Security and Cryptography for Networks*, pages 21–38, September 2008.

[JSLM06a] Yves Igor Jerschow, Björn Scheuermann, Christian Lochert, and Martin Mauve. A cross-layer protocol evaluation framework on esb nodes. In *REALMAN '06: Proceedings of the 2nd International Workshop on Multi-hop Ad Hoc Networks: from Theory to Reality*, pages 104–106, May 2006. Demo Session.

[JSLM06b] Yves Igor Jerschow, Björn Scheuermann, Christian Lochert, and Martin Mauve. A real-world framework to evaluate cross-layer protocols for wireless multihop networks. In *REALMAN '06: Proceedings of the 2nd International Workshop on Multi-hop Ad Hoc Networks: from Theory to Reality*, pages 1–6, May 2006.

[LCS+05] Christian Lochert, Murat Caliskan, Björn Scheuermann, Andreas Barthels, Alfonso Cervantes, and Martin Mauve. Multiple Simulator Interlinking Environment for Inter Vehicle Communication. In *VANET '05: Proceedings of the 2nd ACM International Workshop on Vehicular Ad Hoc Networks*, pages 87–88, September 2005.

[LHT+03] Christian Lochert, Hannes Hartenstein, Jing Tian, Dagmar Herrmann, Holger Füßler, and Martin Mauve. A routing strategy for vehicular ad hoc networks in city environments. In *IV '03: Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 156–161, June 2003.

[LMFH04] Christian Lochert, Martin Mauve, Holger Füßler, and Hannes Hartenstein. Geographic Routing in City Scenarios. In *MobiCom '04: Proceedings of the 10th Annual ACM International Conference on Mobile Computing and Networking*, September 2004. Poster Session.

[LMFH05] Christian Lochert, Martin Mauve, Holger Füßler, and Hannes Hartenstein. Geographic Routing in City Scenarios. *ACM SIGMOBILE Mobile Computing and Communications Review (MC2R)*, 9(1):69–72, January 2005.

[Loc08] Christian Lochert. *Ad-Hoc-Netzwerke in der Fahrzeug-Fahrzeug-Kommunikation: Positionsbasiertes Routing in Städten.* VDM Verlag Dr. Müller, Saarbrücken, Germany, May 2008. In German.

[LRSM08] Christian Lochert, Jedrzej Rybicki, Björn Scheuermann, and Martin Mauve. Scalable data dissemination for inter-vehicle-communication: Aggregation versus peer-to-peer. *Oldenbourg it – Information Technology*, 50(4):237–242, August 2008.

[LSCM07] Christian Lochert, Björn Scheuermann, Murat Caliskan, and Martin Mauve. The feasibility of information dissemination in vehicular ad-hoc networks. In *WONS '07: Proceedings of the 4th Annual Conference on Wireless On-demand Network Systems and Services*, pages 92–99, January 2007.

[LSMa] Christian Lochert, Björn Scheuermann, and Martin Mauve. Obstacle modeling for VANETs. In preparation.

[LSMb] Christian Lochert, Björn Scheuermann, and Martin Mauve. A probabilistic method for cooperative hierarchical aggregation of data in VANETs. *Elsevier Ad Hoc Networks*. Submitted for publication.

[LSM07a] Christian Lochert, Björn Scheuermann, and Martin Mauve. A Survey on Congestion Control for Mobile Ad-Hoc Networks. *Wiley Wireless Communications and Mobile Computing*, 7(5):655–676, June 2007.

[LSM07b] Christian Lochert, Björn Scheuermann, and Martin Mauve. Probabilistic aggregation for data dissemination in VANETs. In *VANET '07: Proceedings of the 4th ACM International Workshop on Vehicular Ad Hoc Networks*, pages 1–8, September 2007.

[LSM09] Christian Lochert, Björn Scheuermann, and Martin Mauve. Information dissemination in VANETs. In Hannes Hartenstein and Kenneth P. Laberteaux, editors, *Vehicular Applications and Inter-Networking Technologies (VANETs)*. Wiley, 2009. To appear.

[LSW+08] Christian Lochert, Björn Scheuermann, Christian Wewetzer, Andreas Luebke, and Martin Mauve. Data aggregation and roadside unit placement for a VANET traffic information system. In *VANET '08: Proceedings of the 5th ACM International Workshop on VehiculAr Inter-NETworking*, pages 58–65, September 2008.

[RSK+07] Jedrzej Rybicki, Björn Scheuermann, Wolfgang Kiess, Christian Lochert, Pezhman Fallahi, and Martin Mauve. Challenge: Peers on wheels – a road to new traffic information systems. In *MobiCom '07: Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking*, pages 215–221, September 2007.

[SKLM] Björn Scheuermann, Markus Koegel, Christian Lochert, and Martin Mauve. Reliable wireless multihop communication without end-to-end acknowledgments. In preparation.

[SLM08a] Björn Scheuermann, Christian Lochert, and Martin Mauve. Implicit hop-by-hop congestion control in wireless multihop networks. *Elsevier Ad Hoc Networks*, 6(2):260–286, April 2008.

[SLM08b] Björn Scheuermann, Christian Lochert, and Martin Mauve. Kooperative schichtenübergreifende Überlastkontrolle für mobile Ad-Hoc-Netzwerke. In Martina Zitterbart, editor, *Basissoftware für Selbstorganisierende Infrastrukturen für Vernetzte Mobile Systeme*, pages 17–32. 2008. In German. To appear.

[SLRM] Björn Scheuermann, Christian Lochert, Jedrzej Rybicki, and Martin Mauve. Limitations and opportunities of data aggregation in VANETs. Submitted for publication.

[STL+07] Björn Scheuermann, Matthias Transier, Christian Lochert, Martin Mauve, and Wolfgang Effelsberg. Backpressure multicast congestion control in mobile ad-hoc networks. In *CoNEXT '07: Proceedings of the 3rd International Conference on Emerging Networking Experiments and Technologies*, pages 265–275, December 2007.

Other References

[AEK+06] Christian Adler, Stephan Eichler, Timo Kosch, Christoph Schroth, and Markus Strassberger. Self-organized and context-adaptive information diffusion in vehicular ad hoc networks. In *ISWCS '06: Proceedings of the 3rd International Symposium on Wireless Communication Systems*, pages 307–311, September 2006.

[AESS06] Christian Adler, Robert Eigner, Christoph Schroth, and Markus Strassberger. Context-adaptive information diffusion in VANETs: Maximizing the global benefit. In *CSN '06: Proceedings of the 5th IASTED International Conference on Communication Systems and Networks*, August 2006.

[ASL07] Ashish Agarwal, David Starobinski, and Thomas D. C. Little. Exploiting downstream mobility to achieve fast upstream message propagation in vehicular ad hoc networks. In *MOVE '07: Proceedings of the 2007 Workshop on Mobile Networking for Vehicular Environments*, pages 13–18, May 2007.

[ASL08] Ashish Agarwal, David Starobinski, and Thomas D. C. Little. Analytical model for message propagation in delay tolerant vehicular ad hoc networks. In *VTC '08-Spring: Proceedings of the 67th IEEE Vehicular Technology Conference*, pages 3067–3071, May 2008.

[BCTL08] Nilanjan Banerjee, Mark D. Corner, Don Towsley, and Brian Neil Levine. Relays, base stations, and meshes: Enhancing mobile networks with infrastructure. In *MobiCom '08: Proceedings of the 14th Annual ACM International Conference on Mobile Computing and Networking*, September 2008.

[BFSS07] Holger Bast, Stefan Funke, Peter Sanders, and Dominik Schultes. Fast routing in road networks with transit nodes. *Science*, 316(5824):566, April 2007.

[BHK05] Jeppe Brønsted, Klaus Marius Hansen, and Lars Michael Kristensen. An infrastructure for a traffic warning system. In *ICPS '05: Proceedings of the International Conference on Pervasive Services*, pages 136–145, July 2005.

[bis] Bison - gnu parser generator. `http://www.gnu.org/software/bison/`.

[BK06] Jeppe Brønsted and Lars Michael Kristensen. Specification and performance evaluation of two zone dissemination protocols for vehicular ad-hoc networks. In *ANSS '06: Proceedings of the 39th Annual Simulation Symposium*, page 12, April 2006.

[BL02] Prithwish Basu and Thomas D.C. Little. Networked parking spaces: architecture and applications. In *VTC '02-Fall: Proceedings of the 56th IEEE Vehicular Technology Conference*, volume 2, pages 1153–1157, September 2002.

[BL04] Prithwish Basu and Thomas D. C. Little. Wireless ad hoc discovery of parking meters. In *WAMES '04: Proceedings of the MobiSys 2004 Workshop on Applications of Mobile Embedded Systems*, pages 8–11, June 2004.

[BM07] Anna Bialas-Motyl. Straßenverkehrssicherheit in der EU im Jahr 2004: Regionale Unterschiede. *EUROSTAT Statistik kurz gefasst*, 2007(14), February 2007.

[BONL08] Azzedine Boukerche, Horacio A. B. F. Oliveira, Eduardo F. Nakamura, and Antonio A. F. Loureiro. Vehicular ad hoc networks: A new challenge for localization-based systems. *Elsevier Computer Communications*, 31:2838–2849, July 2008.

[Bow64] Robert Wayne Bowes. *A study of the feasibility of using roadside radio communications for traffic control and driver information*. PhD thesis, Georgia Institute of Technology, Atlanta, GA, USA, May 1964.

[BSH00] Linda Briesemeister, Lorenz Schäfers, and Günter Hommel. Disseminating messages among highly mobile hosts based on inter-vehicle communication. In *IV '00: Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 522–527, October 2000.

[Bun08] Statistisches Bundesamt. Unfallgeschehen im Straßenverkehr 2007. Fachserie 5462401089004, Statistisches Bundesamt Deutschland, Wiesbaden, July 2008. In German.

[C2C] Car 2 Car Communication Consortium. http://www.car-2-car.org/.

[CB64] Donald O. Covault and Robert Wayne Bowes. A study of the feasibility of using roadside radio communications for traffic control and driver information. *Highway Research Record*, 49:89–106, 1964.

[CBSM07] Murat Caliskan, Andreas Barthels, Björn Scheuermann, and Martin Mauve. Predicting parking lot occupancy in vehicular ad-hoc networks. In *VTC '07-Spring: Proceedings of the 65th IEEE Vehicular Technology Conference*, pages 277–281, April 2007.

[CGM06] Murat Caliskan, Daniel Graupner, and Martin Mauve. Decentralized discovery of free parking places. In *VANET '06: Proceedings of the 3rd ACM International Workshop on Vehicular Ad Hoc Networks*, pages 30–39, September 2006.

[CLC08] Wang-Rong Chang, Hui-Tang Lin, and Bo-Xuan Chen. Trafficgather: An efficient and scalable data collection protocol for vehicular ad hoc networks. In *CCNC '08: Proceedings of the 5th IEEE Consumer Communications and Networking Conference*, pages 365–369, January 2008.

[CLKB04] Jeffrey Considine, Feifei Li, George Kollios, and John Byers. Approximate aggregation techniques for sensor databases. In *ICDE '04: Proceedings of the 20th International Conference on Data Engineering*, pages 449–460, March 2004.

[CM08] Kevin Collins and Gabriel-Miro Muntean. A vehicle route management solution enabled by wireless vehicular networks. In *MOVE '08: Proceedings of the 2008 Workshop on Mobile Networking for Vehicular Environments*, pages 1–6, April 2008.

[COR] The microscopic traffic simulator CORSIM. http://mctrans.ce.ufl.edu/featured/tsis/Version5/corsim.htm.

[CP95] Erick Cantú-Paz. A summary of research on parallel genetic algorithms. IlliGAL Report 95007, University of Illinois at Urbana-Champaign, July 1995.

[CPSdAGB98] Manuel F. Catedra, Jesús Pérez, Francisco Saez de Adana, and Oscar Gutiérrez Blanco. Efficient ray-tracing techniques for three-dimensional analyses of propagation in mobile communications: application to picocell and microcell scenarios. *IEEE Antennas and Propagation Magazine*, 40(2):15–28, April 1998.

[CS04] Ioan Chisalita and Nahid Shahmehri. A context-based vehicular communication protocol. In *PIMRC '04: Proceedings of the 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, volume 4, pages 2820–2824, September 2004.

[CVCAC06] Jasmine Chennikara-Varghese, Wai Chen, Onur Altintas, and Shengwei Cai. Survey of routing protocols for inter-vehicle communications. In *MobiQuitous '06: Proceedings of the Third Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services – Workshops*, pages 1–5, July 2006.

[CVCHO07] Jasmine Chennikara-Varghese, Wai Chen, Toshiro Hikita, and Ryokichi Onishi. Local peer groups and vehicle-to-infrastructure communications. In *GLOBECOM '07: Proceedings of the IEEE Global Telecommunications Conference – Workshops*, pages 1–6, November 2007.

[Deu08] Statistisches Bundesamt Deutschland. Unfälle und Verunglückte im Straßenverkehr. http://tinyurl.com/6aqv7a, 2008. In German.

[DF03] Marianne Durand and Philippe Flajolet. Loglog counting of large cardinalities. In *ESA '03: Proceedings of the 11th Annual European Symposium on Algorithms*, pages 605–617, September 2003.

[DJ07] Sandor Dornbush and Anupam Joshi. StreetSmart Traffic: Discovering and disseminating automobile congestion using vanet's. In *VTC '07-Spring: Proceedings of the 65th IEEE Vehicular Technology Conference*, pages 11–15, April 2007.

[DKS07] Bertrand Ducourthial, Yacine Khaled, and Mohamed Shawky. Conditional transmissions: Performance study of a new communication strategy in VANET. *IEEE Transactions on Vehicular Technology*, 56(6):3348–3357, November 2007.

[DND+04] Sasan Dashtinezhad, Tamer Nadeem, Bogdan Dorohonceanu, Cristian Borcea, Porlin Kang, and Liviu Iftode. Trafficview: a driver assistant device for traffic monitoring based on car-to-car communication. In *VTC '04-Spring: Proceedings of the 59th IEEE Vehicular Technology Conference*, volume 5, pages 2946–2950, May 2004.

[DSR] The DSRC project. http://www.leearmstrong.org/DSRC/-DSRCHomeset.htm.

[DW07] Daniel Delling and Dorothea Wagner. Landmark-based routing in dynamic graphs. In *WEA '07: Proceedings of the 6th Workshop on Experimental Algorithms*, pages 52–65, June 2007.

[Enk03] Wilfried Enkelmann. FleetNet – Applications for Inter-Vehicle Communication. In *IV '03: Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 162–167, June 2003.

[FFGM07] Philippe Flajolet, Éric Fusy, Olivier Gandouet, and Frédéric Meunier. Hyperloglog: The analysis of a near-optimal cardinality estimation algorithm. In *AofA '07: Proceedings of the 2007 International Conference on Analysis of Algorithms*, June 2007.

[FKUH07] Takeshi Fujiki, Masayuki Kirimura, Takaaki Umedu, and Teruo Higashino. Efficient acquisition of local traffic information using inter-vehicle communication with queries. In *ITSC '07: Proceedings of the 10th International IEEE Conference on Intelligent Transportation Systems*, pages 241–246, September 2007.

[fle] flex: The fast lexical analyzer. `http://flex.sourceforge.net/`.

[FM85] Philippe Flajolet and G. Nigel Martin. Probabilistic counting algorithms for data base applications. *Journal of Computer and System Sciences*, 31(2):182–209, October 1985.

[GBMY97] David J. Goodman, Joan Borràs, Narayan B. Mandayam, and Roy D. Yates. INFOSTATIONS: a new system model for data and messaging services. In *VTC '97: Proceedings of the 47th IEEE Vehicular Technology Conference*, volume 2, pages 969–973, May 1997.

[GGS04] Philippe Golle, Dan Greene, and Jessica Staddon. Detecting and correcting malicious data in VANETs. In *VANET '04: Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks*, pages 29–37, October 2004.

[GHL+98] David N. Gray, John Hotchkiss, Seth LaForge, Andrew Shalit, and Toby Weinberg. Modern languages and microsoft's component object model. *Communications of the ACM*, 41(5):55–65, May 1998.

[GIO04] Samir Goel, Tomasz Imielinski, and Kaan Ozbay. Ascertaining viability of WiFi based vehicle-to-vehicle network for traffic information dissemination. In *ITSC '04: Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems*, pages 1086–1091, October 2004.

[GK00] Piyush Gupta and Panganamala R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, March 2000.

[Gol89] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, January 1989.

[HA04] Uwe Hatnik and Sven Altmann. Using ModelSim, Matlab/Simulink and NS for simulation of distributed systems. In *PARELEC '04: Proceedings of the Parallel Computing in Electrical Engineering, International Conference on (PARELEC'04)*, pages 114–119, Washington, DC, USA, September 2004. IEEE Computer Society.

[HM95] Dirk Helbing and Péter Molnár. Social force model for pedestrian dynamics. *American Physical Society Physical Review E*, 51(5):4282–4286, May 1995.

[HNR68] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, July 1968.

[HNR72] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. Correction to "a formal basis for the heuristic determination of minimum cost paths". *ACM SIGART Bulletin*, (37):28–29, December 1972.

[IEE07] IEEE standard for information technology-telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements – part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999)*, pages C1–1184, December 2007.

[IEE08] Task group p - wireless access for the vehicular environment (802.11p amendment). *IEEE Std 802.11p Draft 3.03*, January 2008.

[IFTK00] Shinichi Ichitsubo, Tatsuo Furuno, Tokio Taga, and Ryoji Kawasaki. Multipath propagation model for line-of-sight street microcells in urban area. *Vehicular Technology, IEEE Transactions on*, 49(2):422–427, March 2000.

[Int03] International Organization for Standardization. Road vehicles – controller area network (CAN). *ISO 11898*, November 2003.

[Int04] International Organization for Standardization. Road vehicles – diagnostics on controller area networks (CAN). *ISO 11765*, October 2004.

[ITS] Intelligent transportation systems. http://www.its.dot.gov/index.htm.

[IW07] Khaled Ibrahim and Michele C. Weigle. Accurate data aggregation for VANETs. In *VANET '07: Proceedings of the 4th ACM International Workshop on Vehicular Ad Hoc Networks*, pages 71–72, September 2007. Poster Session.

[IW08a] Khaled Ibrahim and Michele C. Weigle. Optimizing CASCADE data aggregation for VANETs. In *MoVeNet '08: Proceedings of the 2nd International Workshop on Mobile Vehicular Networks*, September 2008.

[IW08b] Khaled Ibrahim and Michele C. Weigle. Securing CASCADE data aggregation for VANETs. In *MoVeNet '08: Proceedings of the 2nd International Workshop on Mobile Vehicular Networks*, September 2008. Poster Session.

[JSRGD07] Moez Jerbi, Sidi-Mohammed Senouci, Tinku Rasheed, and Yacine Ghamri-Doudane. An infrastructure-free traffic information system for vehicular networks. In *VTC '07-Fall: Proceedings of the 66th IEEE Vehicular Technology Conference*, pages 2086–2090, October 2007.

[JW04] Markus Jakobsson and Susanne Wetzel. Efficient attribute authentication with applications to ad hoc networks. In *VANET '04: Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks*, pages 38–46, October 2004.

[KEÖÖ04] Gökhan Korkmaz, Eylem Ekici, Füsun Özgüner, and Ümit Özgüner. Urban multi-hop broadcast protocol for inter-vehicle communication systems. In *VANET '04: Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks*, pages 76–85, October 2004.

[KGJWI$^+$99] Seong-Cheol Kim, Bernard J. Guarino Jr., Thomas M. Willis III, Vinko Erceg, Steven J. Fortune, Reinaldo A. Valenzuela, Louis W. Thomas, Jonathan Ling, and J. Don Moore. Radio propagation measurements and prediction using three-dimensional ray tracing in urban environments at 908 MHz and 1.9 GHz. *IEEE Transactions on Vehicular Technology*, 48(3):931–946, May 1999.

[KHRW02] Daniel Krajzewicz, Georg Hertkorn, Christian Rössel, and Peter Wagner. SUMO (simulation of urban mobility): An open-source traffic simulation. In *MESM '02: Proceedings of the 4th Middle East Symposium on Simulation and Modelling*, pages 183–187, September 2002.

[KK00] Brad Karp and Hsiang-Tsung Kung. GPSR: greedy perimeter stateless routing for wireless networks. In *MobiCom '00: Proceedings of the 6th Annual ACM International Conference on Mobile Computing and Networking*, pages 243–254, August 2000.

[Knu64] Donald E. Knuth. Backus normal form vs. Backus Naur form. *Communications of the ACM*, 7(12):735–736, 1964.

[KPS92] Peter Kirschenhofer, Helmut Prodinger, and Wojciech Szpankowski. How to count quickly and accurately: A unified analysis of probabilistic counting and other related problems. In *ICALP '92: Proceedings of the 19th International Colloquium on Automata, Languages and Programming*, pages 211–222, July 1992.

[KSA02] Timo Kosch, Christian Schwingenschlögl, and Li Ai. Information dissemination in multihop inter-vehicle networks. In *ITSC '02: Proceedings of the 5th International IEEE Conference on Intelligent Transportation Systems*, pages 685–690, September 2002.

[KWG97] Stefan Krauss, Peter Wagner, and Christian Gawron. Metastable states in a microscopic model of traffic flow. *American Physical Society Physical Review E*, 55(5):5597–5602, May 1997.

[KY08] Zhenning Kong and Edmund M. Yeh. On the latency for information dissemination in mobile wireless networks. In *MobiHoc '08: Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 139–148, May 2008.

[LA05] Thomas D. C. Little and Ashish Agarwal. An information propagation scheme for VANETs. In *ITSC '05: Proceedings of the 8th International IEEE Conference on Intelligent Transportation Systems*, pages 155–160, September 2005.

[LB98] George Liang and Henry L. Bertoni. A new approach to 3-D ray tracing for propagation prediction in cities. *IEEE Transactions on Antennas and Propagation*, 46(6):853–863, June 1998.

[LIW] The LIWAS project. http://www.liwas.dk/gb/.

[LM07] Ilias Leontiadis and Cecilia Mascolo. Opportunistic spatio-temporal dissemination system for vehicular networks. In *MobiOpp '07: Proceedings of the 1st international MobiSys workshop on Mobile opportunistic networking*, pages 39–46, June 2007.

[LW07] Fan Li and Yu Wang. Routing in vehicular ad hoc networks: A survey. *IEEE Vehicular Technology Magazine*, 2(2):12–22, June 2007.

[Mar08] John Markoff. Can't find a parking spot? check smartphone. *The New York Times*, (12), July 2008.

[Mat] The Matlab/Simulink application simulator. http://www.mathworks.com/ products/simulink/.

[MMP+05] Kirsten Matheus, Rolf Morich, Ingrid Paulus, Cornelius Menig, Andreas Luebke, Bernd Rech, and Will Specks. Car-to-car communication – market introduction and success factors. In *ITS '05: Proceedings of the 5th European Congress and Exhibition on Intelligent Transport Systems and Services*, June 2005.

[MWH01] Martin Mauve, Jörg Widmer, and Hannes Hartenstein. A survey on position-based routing in mobile ad-hoc networks. *IEEE Network Magazine*, 15(6):30–39, November 2001.

[NBB+08] Giacomo Nannicini, Philippe Baptiste, Gilles Barbier, Daniel Krob, and Leo Liberti. Fast paths in large-scale dynamic road networks. *Springer Computational Optimization and Applications*, 2008. Online preprint.

[NDLI04a] Tamer Nadeem, Sasan Dashtinezhad, Chunyuan Liao, and Liviu Iftode. TrafficView: A scalable traffic monitoring system. In *MDM '04: Proceedings of the 5th IEEE International Conference on Mobile Data Management*, pages 13–26, January 2004.

[NDLI04b] Tamer Nadeem, Sasan Dashtinezhad, Chunyuan Liao, and Liviu Iftode. TrafficView: traffic data dissemination using car-to-car communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 8(3):6–19, July 2004.

[NoW] The Network-on-Wheels project. http://www.network-on-wheels.de.

[ns2] The network simulator ns-2. http://www.isi.edu/nsnam/ns/.

[NSI06] Tamer Nadeem, Pravin Shankar, and Liviu Iftode. A comparative study of data dissemination models for VANETs. In *MobiQuitous '06: Proceedings of the Third Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services*, pages 1–10, July 2006.

[NTCS99] Sze-Yao Ni, Yu-Chee Tseng, Yuh-Shyan Chen, and Jang-Ping Sheu. The broadcast storm problem in a mobile ad hoc network. In *MobiCom '99: Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 151–162, July 1999.

[ONG07] Kaan Ozbay, Hani Nassif, and Samir Goel. Propagation characteristics of dynamic information collected by in-vehicle sensors in a vehicular network. In *IV '07: Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 1089–1094, June 2007.

[OZRM00] Larry E. Owen, Yunlong Zhang, Lei Rao, and Gene McHale. Street and traffic simulation: traffic flow simulation using CORSIM. In *WSC '00: Proceedings of the 32nd conference on Winter simulation*, pages 1143–1147, San Diego, CA, USA, December 2000. Society for Computer Simulation International.

[PAT] The PATH project. http://www.path.berkeley.edu.

[PGP06] Michal Piorkowski, Matthias Grossglauser, and Aristidis Papaioannou. Mobile user navigation supported by WSAN: Full-fledge demo of the SmartPark system. In *MobiHoc '06: Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, May 2006. Technical Demonstration.

[PR99] Charles E. Perkins and Elizabeth M. Royer. Ad-hoc on-demand distance vector routing. In *WMCSA '99: Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, February 1999.

[PRe] The PReVENT project. http://www.prevent-ip.org.

[PRGI06] Fabio Picconi, Nishkam Ravi, Marco Gruteser, and Liviu Iftode. Probabilistic validation of aggregated data in vehicular ad-hoc networks. In *VANET '06: Proceedings of the 3rd ACM International Workshop on Vehicular Ad Hoc Networks*, pages 76–85, September 2006.

[PTV] PTV AG. The traffic simulator VISSIM. http://www.english.ptv.de/cgi-bin/traffic/traf_vissim.pl.

[PZ08] Soyoung Park and Cliff C. Zou. Reliable traffic information propagation in vehicular ad-hoc networks. In *SARNOFF '08: Proceedings of the IEEE Sarnoff Symposium*, pages 1–6, April 2008.

[RAH06] Maxim Raya, Adel Aziz, and Jean-Pierre Hubaux. Efficient secure aggregation in VANETs. In *VANET '06: Proceedings of the 3rd ACM International Workshop on Vehicular Ad Hoc Networks*, pages 67–75, September 2006.

[RL79] Jorma J. Rissanen and Glen G. Langdon. Arithmetic coding. *IBM Journal of Research and Development*, 23(2):149–162, March 1979.

[RML02] Michael Rudack, Michael Meincke, and Matthias Lott. On the dynamics of ad-hoc networks for inter vehicle communications. In *ICWN '02: Proceedings of the International Conference on Wireless Networks*, June 2002.

[RWG97] Karim Rizk, Jean-Frédéric Wagen, and Fred Gardiol. Two-dimensional ray-tracing modeling for propagation prediction in microcellular environments. *IEEE Transactions on Vehicular Technology*, 46(2):508–518, May 1997.

[SD07] S. N. Sivanandam and S. N. Deepa. *Introduction to Genetic Algorithms*. Springer Verlag, Berlin, Germany, 1 edition, December 2007.

[SFUH04] Masashi Saito, Mayuko Funai, Takaaki Umedu, and Teruo Higashino. Inter-vehicle ad-hoc communication protocol for acquiring local traffic information. In *Proceedings of the 11th World Congress on Intelligent Transport Systems*, October 2004.

[SHR05] Illya Stepanov, Daniel Herrscher, and Kurt Rothermel. On the impact of radio propagation models on MANET simulation results. In *MWCN '05: Proceedings of the 7th IFIP International Conference on Mobile and Wireless Communication Networks*, September 2005.

[SJK+03] Tapan K. Sarkar, Zhong Ji, Kyungjung Kim, Abedellatif Medouri, and Magdalena Salazar-Palma. A survey of various propagation models for mobile communication. *IEEE Antennas and Propagation Magazine*, 45(3):51–82, June 2003.

[SM07] Björn Scheuermann and Martin Mauve. Near-Optimal Compression of Probabilistic Counting Sketches for Networking Applications. In *Dial M-POMC 2007: Proceedings of the 4th ACM SIGACT-SIGOPS International Workshop on Foundation of Mobile Computing*, August 2007.

[SMZ+02] Abdullah E. Shaikh, Faisal Majeed, Muhammad Zeeshan, Tahir Rabbani, and Imran Sheikh. Efficient implementation of deterministic 3-D ray tracing model to predict propagation losses in indoor environments. In *PIMRC '02: Proceedings of the 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, volume 3, pages 1208–1212, September 2002.

[SS06] Peter Sanders and Dominik Schultes. Engineering highway hierarchies. In *Proceedings of the 14th Annual European Symposium*, volume 4168, pages 804–816. Springer, September 2006.

[SSHN07] Hiroyuki Sago, Masako Shinohara, Takahiro Hara, and Shojiro Nishio. A data dissemination method for information sharing based on inter-vehicle communication. In *AINAW '07: Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops*, volume 2, pages 743–748, May 2007.

[STC+06] Davide Sormani, Gabriele Turconi, Paolo Costa, Davide Frey, Matteo Migliavacca, and Luca Mottola. Towards lightweight information dissemination in inter-vehicular networks. In *VANET '06: Proceedings of the 3rd ACM International Workshop on Vehicular Ad Hoc Networks*, pages 20–29, September 2006.

[STK+06] Takashi Shinkawa, Takashi Terauchi, Tomoya Kitani, Naoki Shibata, Keiichi Yasumoto, Minoru Ito, and Teruo Higashino. A technique for information sharing using inter-vehicle communication with message ferrying. In *MDM '06: Proceedings of the 7th IEEE International Conference on Mobile Data Management*, pages 130–134, May 2006.

[Sto02] Ivan Stojmenovic. Position-based routing in ad hoc networks. *IEEE Communications Magazine*, 40(7):128–134, July 2002.

[STUH07] Masashi Saito, Jun Tsukamoto, Takaaki Umedu, and Teruo Higashino. Design and evaluation of intervehicle dissemination protocol for propagation of preceding traffic information. *IEEE Transactions on Intelligent Transportation Systems*, 8(3):379–390, September 2007.

[SUM] The microscopic traffic simulator SUMO. http://sumo.sourceforge.net/index.shtml.

[TKC+04] Yufei Tao, George Kollios, Jeffrey Considine, Feifei Li, and Dimitris Papadias. Spatio-temporal aggregation using sketches. In *ICDE '04: Proceedings of the 20th International Conference on Data Engineering*, pages 214–226, March 2004.

[TM07] Marc Torrent-Moreno. Inter-vehicle communications: assessing information dissemination under safety constraints. In *WONS '07: Proceedings of the 4th Annual Conference on Wireless On-demand Network Systems and Services*, pages 59–64, January 2007.

[TMJH04] Marc Torrent-Moreno, Daniel Jiang, and Hannes Hartenstein. Broadcast reception rates and effects of priority access in 802.11-based vehicular adhoc networks. In *VANET '04: Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks*, pages 10–18, October 2004.

[TMSH05] Marc Torrent-Moreno, Paolo Santi, and Hannes Hartenstein. Fair sharing of bandwidth in vanets. In *VANET '05: Proceedings of the 2nd ACM International Workshop on Vehicular Ad Hoc Networks*, pages 49–58, September 2005.

[TMSH06] Marc Torrent-Moreno, Paolo Santi, and Hannes Hartenstein. Distributed fair transmit power adjustment for vehicular ad hoc networks. In *SECON '06: Proceedings of the 3rd Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, volume 2, pages 479–488, September 2006.

[TT95] S.Y. Tan and H.S. Tan. Propagation model for microcellular communications applied to path loss measurements in Ottawa city streets. *IEEE Transactions on Vehicular Technology*, 44(2):313–317, May 1995.

[TWB+07] Ozan Tonguz, Nawaporn Wisitpongphan, Fan Bai, Priyantha Mudalige, and Varsha Sadekar. Broadcasting in VANET. In *MOVE '07: Proceedings of the 2007 Workshop on Mobile Networking for Vehicular Environments*, pages 7–12, May 2007.

[Val03] Reinaldo A. Valenzuela. A ray tracing approach to predicting indoor wireless transmission. In *VTC '93: Proceedings of the 43rd IEEE Vehicular Technology Conference*, pages 214–218, May 2003.

[WC02] Brad Williams and Tracy Camp. Comparison of broadcasting techniques for mobile ad hoc networks. In *MobiHoc '02: Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 194–205, June 2002.

[WER+03a] Lars Wischhof, André Ebner, Hermann Rohling, Matthias Lott, and Rüdiger Halfmann. Adaptive broadcast for travel and traffic information distribution based on inter-vehicle communication. In *IV '03: Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 6–11, June 2003.

[WER+03b] Lars Wischhof, André Ebner, Hermann Rohling, Matthias Lott, and Rüdiger Halfmann. SOTIS – a self-organizing traffic information system. In *VTC '03-Spring: Proceedings of the 57th IEEE Vehicular Technology Conference*, pages 2442–2446, April 2003.

[WER05] Lars Wischhof, André Ebner, and Hermann Rohling. Information dissemination in self-organizing intervehicle networks. *IEEE Transactions on Intelligent Transportation Systems*, 6(1):90–101, March 2005.

[WFGH04] Hao Wu, Richard Fujimoto, Randall Guensler, and Michael Hunter. MDDV: A mobility-centric data dissemination algorithm for vehicular networks. In *VANET '04: Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks*, pages 47–56, October 2004.

[WHF+07] Axel Wegener, Horst Hellbrück, Stefan Fischer, Christiane Schmidt, and Sándor Fekete. AutoCast: An adaptive data dissemination protocol for traffic information systems. In *VTC '07-Fall: Proceedings of the 66th IEEE Vehicular Technology Conference*, pages 1947–1951, October 2007.

[Wie74] Rainer Wiedemann. Simulation des Straßenverkehrsflusses. *Schriftenreihe des Instituts für Verkehrswesen der Universität Karlsruhe*, 8, 1974. In German.

[WNC87] Ian H. Witten, Radford M. Neal, and John G. Cleary. Arithmetic coding for data compression. *Communications of the ACM*, 30(6):520–540, June 1987.

[XB06a] Huaying Xu and Matthew Barth. An adaptive dissemination mechanism for intervehicle communication-based decentralized traffic information systems. In *ITSC '06: Proceedings of the 9th International IEEE Conference on Intelligent Transportation Systems*, pages 1207–1213, September 2006.

[XB06b] Huaying Xu and Matthew Barth. Travel time estimation techniques for traffic information systems based on inter-vehicle communications. In *TRB '06: 82nd Annual Transportation Research Board Meeting*, January 2006.

[XMKS04] Qing Xu, Tony Mak, Jeff Ko, and Raja Sengupta. Vehicle-to-vehicle safety messaging in DSRC. In *VANET '04: Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks*, pages 19–28, October 2004.

[XOW04] Bo Xu, Aris Ouksel, and Ouri Wolfson. Opportunistic resource exchange in inter-vehicle ad-hoc networks. In *MDM '04: Proceedings of the 5th IEEE International Conference on Mobile Data Management*, pages 4–12, January 2004.

[YR05] Xu Yang and Will Recker. Simulation studies of information propagation in a self-organizing distributed traffic information system. *Elsevier Transportation Research Part C*, 13(5-6):370–390, October 2005.

[ZC06] Jing Zhao and Guohong Cao. VADD: Vehicle-assisted data delivery in vehicular ad hoc networks. In *INFOCOM '06: Proceedings of the 25th Annual Joint Conference of the IEEE Computer and Communications Societies*, pages 1–12, April 2006.

[ZC08] Jing Zhao and Guohong Cao. VADD: Vehicle-assisted data delivery in vehicular ad hoc networks. *IEEE Transactions on Vehicular Technology*, 57(3):1910–1922, May 2008.

[ZZC07] Jing Zhao, Yang Zhang, and Guohong Cao. Data pouring and buffering on the road: A new data dissemination paradigm for vehicular ad hoc networks. *IEEE Transactions on Vehicular Technology*, 56(6):3266–3277, November 2007.

Index