Inter-Vehicular Communication Simulation
based on Cellular Network Traces

Inter-Vehicular Communication Simulation based on Cellular Network Traces

Inaugural-Dissertation

zur Erlangung des Doktorgrades der Mathematisch-Naturwissenschaftlichen Fakultät der Heinrich-Heine-Universität Düsseldorf

vorgelegt von
Norbert Goebel
aus Neuss

Düsseldorf, Dezember 2017

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

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

Berichterstatter:

1. Prof. Dr. Martin Mauve
   Heinrich-Heine-Universität Düsseldorf
2. Jun-Prof. Dr.-Ing. Kalman Graffi
   Heinrich-Heine-Universität Düsseldorf

Tag der mündlichen Prüfung: 01.02.2018

Abstract

Currently, two competing wireless communication technologies for *inter-vehicular communication* (IVC)[1] exist: IEEE 802.11p *Wireless Local Area Network* (Wi-Fi) and cellular network communication. Both technologies feature different network characteristics like diverse available data rates, delays or drop rates. Often it is not clear which communication technology best fits for a specific IVC application. Even the decision if an IVC application is feasible using Wi-Fi or cellular networks is not simple. While *Field Operational Tests* (FOTs) might be able to answer these questions, they are expensive and time consuming. Thus a simulative approach is desirable to answer these questions and speed up IVC application development. While multiple simulators for IEEE 802.11p exist the situation differs for cellular network simulations as their complexity and diversity make it hard to simulate every aspect of a cellular network. Instead of simulating every tiny aspect of the cellular network we propose to use a trace-based simulation approach to simulate cellular network communication for *Vehicle-to-X* (V2X) simulations, which solely relies on cellular network measurements conductible by everyone and everywhere.

The goal of this thesis is to evaluate the feasibility of trace-based simulation of cellular networks for V2X simulations, and furthermore to develop, evaluate and release a trace-based cellular network simulation model and couple it with a traffic simulator and a V2X application simulator. Pursuing these goals, a number of research questions in the research fields *network measurements*, *map-matching*, *cellular network simulation* and *coupling of multiple simulators* have to be answered.

First, we analyze current state of the art available data rate measurement algorithms and show that neither of them is able to achieve simultaneous high frequency measurements of available data rates, loss rates and one-way delays. As our trace-based simulation model relies on accurate, position-based measurements of these key network characteristics, we thus develop, evaluate and release our own measurement framework, the *Rate Measurement Framework* (RMF). It uses a heavily modified *bulk traffic* measurement methodology that, amongst others, incorporates multiple feedback loops between the measurement devices and thus can cope with the fast changing network conditions found in cellular networks. Furthermore, we present our map-matching, aggregation and filtering tool chain. It post-processes the network traces gathered with the RMF, map-matches the measurements onto a road network graph derived from *OpenStreetMap* (OSM) mapping data and transforms the results into the

[1]C2X and V2X are synonyms for inter-vehicular communication and are used interchangeably in the literature and in this thesis.

graph-based structure our cellular network simulation model requires.

Following our goal to develop and release a trace-based simulation model for cellular networks, we give an overview on existing cellular network simulators and outline why they are limited to specific scenarios and often require confidential information on the cellular network. Thus we introduce and evaluate our trace-based simulation model first as a prototype and later on as an *Objective Modular Network Testbed in C++* (OMNeT++) module, which allows easy integration of our simulation model into existing simulations using the OMNeT++ framework.

We reach our final goal by incorporating our OMNeT++ module into a fully-fledged coupled simulation environment, *V2X Simulation Runtime Infrastructure* (VSimRTI). We demonstrate the usability of our simulation tool chain by simulating the *Emergency Warning Application* (EWA) defined by the *European Telecommunications Standards Institute* (ETSI) with the help of the simulators *Simulation of Urban Mobility* (SUMO), *V2X application simulator* (VSimRTI_App) and OMNeT++ with our cellular network simulation module. The simulations use network characteristics measured using the RMF, and post-processed by our map-matching tool chain. We evaluate the performance of EWA, which is defined for *Institute of Electrical and Electronics Engineers* (IEEE) 802.11p Wi-Fi use, and implement and evaluate a cellular network optimized version of EWA. By simulating the same scenario now using the enhanced EWA version we underline the importance of simulations for V2X application development and show that the modifications significantly raise the performance of EWA.

In conclusion, we introduce and evaluate our Rate Measurement Framework, offer our tool chain that map-matches the network traces onto an OSM-based road network graph, propose, implement and evaluate our trace-based simulation model as an OMNeT++ module and incorporate it into the simulation framework VSimRTI. We evaluate our measurement methodology, demonstrate the feasibility of trace-based cellular network simulations and determine the usability of our simulation tool chain. Furthermore, we provide the sourcecode of all software developed by us and used in the papers presented in this thesis (measurements, post-processing, map-matching and the OMNeT++ simulation model) and thus enable everyone to use our simulation tool chain and to build upon our work.

Zusammenfassung

Aktuell existieren zwei konkurrierende drahtlose Kommunikationstechnologien für die Fahrzeug-zu-Fahrzeug Kommunikation (IVC[2]): *Institute of Electrical and Electronics Engineers* (IEEE) 802.11p *Wireless Local Area Network* (WLAN) und die Kommunikation über Mobilfunknetze. Dabei unterscheiden sich die Netzwerkcharakteristiken, wie verfügbare Datenrate, Paketverlustrate und Latenzen der beiden Technologien. Oft ist nicht direkt ersichtlich, welche Kommunikationstechnologie am besten für die Umsetzung einer spezifischen *inter-vehicular communication* (IVC) Anwendung geeignet ist. Selbst die Entscheidung, ob eine IVC Anwendung überhaupt sinnvoll über WLAN oder Mobilkommunikation umgesetzt werden kann, ist nicht einfach. Feldtests können diese Fragen möglicherweise beantworten, sind aber teuer und zeitaufwendig. Daher sind Simulationen zur Beantwortung der Fragen sinnvoll und helfen die Anwendungsentwicklung zu beschleunigen. Während mehrere Simulatoren für IEEE 802.11p existieren, ist die Lage bei Mobilfunksimulatoren eine andere, da deren Komplexität und Vielfältigkeit eine Simulation aller Teilaspekte des Mobilfunks erschweren. Anstatt der Simulation aller Details eines Mobilfunknetzes schlagen wir stattdessen einen trace-basierten Simulationsansatz für die Mobilkommunikation für *Vehicle-to-X* (V2X) Simulationen vor, der einzig auf Netzwerkmessungen des Mobilfunknetzes aufbaut, die jeder überall durchführen kann.

Das Ziel dieser Dissertation ist zunächst die Machbarkeit einer trace-basierten Simulation von Mobilfunk für V2X Simulationen zu überprüfen. Darüber hinaus soll ein trace-basiertes Mobilfunksimulationsmodell entwickelt und evaluiert werden und mit einem Verkehrssimulator und einem V2X Anwendungssimulator gekoppelt werden. Um diese Ziele zu erreichen müssen eine Vielzahl an Forschungsfragen aus den Bereichen *Netzwerkmessungen*, *Map-Matching*, *Simulation von Mobilfunk* und *Kopplung mehrerer Simulatoren* beantwortet werden.

Zunächst analysieren wir aktuelle Messverfahren für verfügbare Datenraten und zeigen, dass keines dieser Messverfahren fähig ist gleichzeitig und mit hoher Messfrequenz Messungen der verfügbaren Datenraten, der Paketverlustraten und der Latenzen durchzuführen. Da unser trace-basiertes Simulationsmodell genaue, positionsbasierte Messungen dieser Kernnetzwerkcharakteristiken benötigt, entwickeln, evaluieren und veröffentlichen wir daher unser eigenes Messframework (*Rate Measurement Framework* (RMF)). Es benutzt ein stark modifiziertes *bulk traffic* Messverfahren, das unter an-

[2]C2X und V2X sind Synonyme für IVC und beschreiben alle Fahrzeug-zu-Fahrzeug bzw. Fahrzeug-zu-Infrastruktur Kommunikation. Sie werden in der Literatur und in dieser Arbeit austauschbar verwendet.

derem mehrere Rückkopplungen zwischen den Messgeräten beinhaltet und daher mit den sich schnell ändernden Netzcharakteristiken in Mobilfunknetzen umgehen kann. Außerdem stellen wir unsere Software zum Map-Matching, der Aggregation und der Filterung dieser Traces vor. Sie verarbeitet die Netzwerkmessungen, die mit dem RMF erstellt werden, projiziert sie auf einen Straßengraphen, den wir aus *OpenStreetMap* (OSM) Kartendaten erzeugen, und überführt die Daten in eine graph-basierte Struktur, die von unserem Mobilfunksimulationsmodell benötigt wird.

Zur Erreichung eines weiteren Ziels der Entwicklung und Veröffentlichung eines trace-basierten Mobilfunksimulationsmodells geben wir einen Überblick über existierende Mobilfunksimulatoren und zeigen, warum diese limitiert auf spezielle Szenarien sind und oft vertrauliche Informationen über das Mobilfunknetz benötigen. Anschließend stellen wir unser trace-basiertes Simulationsmodell zunächst als Prototypen und später als *Objective Modular Network Testbed in C++* (OMNeT++) Modul vor und evaluieren es. Das OMNeT++ Modul kann einfach in bestehende Simulationen integriert werden, die das OMNeT++ Framework benutzen.

Unser finales Ziel erreichen wir durch die Integration unseres OMNeT++ Moduls in die Simulationsumgebung *V2X Simulation Runtime Infrastructure* (VSimRTI), die mehrere Simulatoren koppelt. Durch die Simulation der von der *European Telecommunications Standards Institute* (ETSI) definierten V2X Anwendung *Emergency Warning Application* (EWA) und mit Hilfe der Simulatoren *Simulation of Urban Mobility* (SUMO), *V2X application simulator* (VSimRTI_App) und OMNeT++ mit unserem Modul zur Simulation von Mobilfunk demonstrieren wir die Nutzbarkeit unserer Simulationsumgebung. Dabei benutzen wir Netzwerkcharakteristiken, die mit dem RMF gemessen und mit unseren Softwaretools nachbearbeitet wurden. Weiterhin analysieren wir die Leistung der für IEEE 802.11p WLAN definierten EWA Anwendung und implementieren und evaluieren eine für Mobilfunkverbindungen optimierte EWA Version. Durch eine weitere Simulation des gleichen Szenarios mit der verbesserten EWA Version unterstreichen wir, wie wichtig die Simulation für die V2X Anwendungsentwicklung ist und wir zeigen, dass unsere Modifikationen die Leistung von EWA signifikant steigern.

Zusammenfassend führen wir unser Messframework *RMF* ein und evaluieren es, stellen unsere Software Toolchain zum Map-Matching der Netzwerktraces auf einem OSM-basierten Straßengraphen vor, entwerfen, implementieren und evaluieren unser trace-basiertes Simulationsmodell als OMNeT++ Modul und binden selbes in das Simulationsframework VSimRTI ein. Wir evaluieren unsere Messmethodik, demonstrieren die Machbarkeit von trace-basierter Mobilfunksimulation und zeigen die Nützlichkeit unserer Simulations Toolchain. Darüber hinaus stellen wir den Sourcecode für alle von uns für diese Dissertation entwickelten Programme (Messframework, Post-Processing, Map-Matching und das OMNeT++ Simulationsmodell) zur Verfügung, damit jeder unsere Simulations Toolchain nutzen und auf unseren Arbeiten aufbauen kann.

Acknowledgments

First of all, I would like to thank Björn Scheuermann, who first suggested that I should think about doctoral studies after finishing my Master Thesis, and who recommended me to Martin Mauve when he was looking for a new staff member.

I am even more grateful to my doctoral advisor, Martin Mauve, who offered me the opportunity to start my doctoral studies in the computer networking group at the Heinrich Heine University Düsseldorf. Thank you for your support and the invaluable input and suggestions during discussions.

Later on during my studies, the university changed its doctoral program and Kalman Graffi became my second doctoral advisor. While it was not easy to fulfill the sometimes disjunct visions of Kalman and Martin, I am glad that I had the opportunity to profit from the knowledge of both of them. Furthermore, Kalman's periodic status update requests are partly responsible that I finally submitted my thesis in 2017.

I would like to thank my colleagues Raed Al-Aaridhi, Tobias Amft, Raphael Bialon, Daniel Baselt, Ahmad Reza Cheraghi, Andreas Disterhöft, Tobias Escher, Sabine Freese, Janine Golov, Philipp Hagemeister, Yves Jerschow, Wolfgang Kies, Tobias Krauthoff, Christian Meter, Matthias Radig, Jedrzej Rybicki, Salem Omar Sati, Alexander Schneider, Pascal Siegers, and Thomas Spitzlei for the fantastic working atmosphere that made working fun — most of the times.

Furthermore, I would like to thank all the students that I supervised during my doctoral studies. Especially Tobias Amft, Raphael Bialon, Christian Lange, Daniel Elmo, Andre Ippisch, Franz Kary, Tobias Krauthoff, Adrian Skuballa and Sebastian Wilken, who developed and discussed prototypes for the map-matching software, the measurement framework, the coupled simulation tool chain or worked with me on third party projects.

Special thanks go to Peter Lieven and kamps.de, who hosted one of our measurement servers, to Wolfgang Kies, who helped to find the subject for my doctoral studies, and to the VSimRTI-guys, who offered great support.

And finally, last but by no means least, I am most grateful for the continuous support of my family, particularly to Sabine who masters her work, cares for our children and still lectors my "nerd work" written in "strange" English.

Contents

List of Figures

Abbreviations

ABS	Anti-lock Braking System
ACC	Adaptive Cruise Control
AMC	Adaptive Modulation and Coding
BLER	Block Error Rate
BS	Base Station
C2X	Car-to-X
CAM	Cooperative Awareness Message
CNS	Cellular Network Simulator
CQI	Channel Quality Indicator
D2D	device-to-device communication
DENM	Decentralized Environmental Notification Message
EBA	Emergency Brake Assist
eNB	Evolved Node B
ESP	Electronic Stability Program
ETSI	European Telecommunications Standards Institute
e.u.r.a.n.e	Enhanced UMTS Radio Access Network Extensions for ns-2
E-UTRA	evolved UMTS Terrestrial Radio Access
EWA	Emergency Warning Application
FDD	Frequency Devision Duplex
FEC	Forward Error Correction
FOT	Field Operational Test
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSoC	Google Summer of Code
GW	Gateway
HARQ	Hybrid Automatic Repeat Request
HS-DSCH	High Speed Downlink Shared Channel
HSDPA	High Speed Downlink Packet Access
HSPA	High Speed Packet Access
HSUPA	High Speed Uplink Packet Access
ICSIC	inter-chirp self-induced congestion

IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
iTETRIS	Integrated Wireless and Traffic Platform for Real-Time Road Traffic Management Solutions
iAPP	iTETRIS Application
iCS	iTETRIS Control System
ITS	Intelligent Transportation Systems
IVC	inter-vehicular communication
JiST/SWANS	Java in Simulation Time / Scalable Wireless Ad hoc Network Simulator
LENA	LTE-EPC Network simulAtor
LTE	Long Term Evolution
LTE-Advanced	Long Term Evolution Advanced
MAC	Medium Access Control
MHT	multiple hypothesis technique
MIMO	Multiple Input and Multiple Output
MiXiM	Mixed Simulator
MME	Mobility Management Entity
NDA	Non-Disclosure Agreement
NED	NEtwork Description
NetSim	Network Simulation and Emulation Platform
ns-2	Network Simulator
ns-3	Network Simulator
OMNeT++	Objective Modular Network Testbed in C++
OSM	OpenStreetMap
OTcl	object-oriented extension to Tcl
PGM	Probe Gap Model
PHY	Physical layer
PRM	Probe Rate Model
RAC	Random Access Channel
RAN	Radio Access Network
RMF	Rate Measurement Framework
RNC	Radio Network Controller
RRC	Radio Resource Control
RRM	Radio Resource Management
SIC	self-induced congestion
sim^{TD}	Sichere Intelligente Mobilität Testfeld Deutschland
SimuLTE	LTE User Plane Simulation Model for INET & OMNeT++
SNR	Signal-to-Noise Ratio
SUMO	Simulation of Urban Mobility

TBUS	Trace Based UMTS Simulation
TraCI	Traffic Control Interface
UE	User Equipment
UMTS	Universal Mobile Telecommunications System
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-X
VANET	Vehicle ad hoc Network
Veins	Vehicles in Network Simulation
VeinsLTE	Vehicles in Network Simulation merged with SimuLTE
VISSIM	Verkehr In Städten - SIMulationsmodell
VSimRTI	V2X Simulation Runtime Infrastructure
VSimRTI_App	V2X application simulator
VSimRTI_Cell	VSimRTI Cellular Network Simulator
WCDMA	Wideband Code Division Multiple Access
Wi-Fi	Wireless Local Area Network
WLAN	Wireless Local Area Network

1 Introduction

The industrial development of the last decades vastly changes the demands on mobility. On the one hand, many employees no longer live in the vicinity of their workplace and thus have to commute on a daily basis. On the other hand, the *just-in-time* production leads to tight constraints on product shipments. Combined with world-wide population growth this leads to a rising number of vehicles on the streets. Consequently, traffic densities are growing, especially road traffic densities. More vehicles sharing the roads lead to increasing numbers of accidents. The *Statistische Bundesamt* for example lists 2.31 million accidents recorded by the German police in 1991, compared to 2.59 million accidents in 2016 [5].

On the one hand, national laws and regulations like (using the example of Germany) speed limits in cities, the obligation to use children's seats, the requirement to wear helmets on motorcycles and better management of traffic flows help to reduce the number of fatal accidents. Naturally, on the other hand, the automotive industry is challenged to develop vehicles that help to avoid accidents or at least reduce their severity. This lead to inventions like *seat belts*, *Anti-lock Braking System* (ABS), *Electronic Stability Program* (ESP), *airbags*, *Adaptive Cruise Control* (ACC) and *Emergency Brake Assist* (EBA). The combined effort of jurisdiction and automotive industry lead, in the same timespan from 1991 to 2016, to a reduction of accidents with injured persons from 516 thousand to 400 thousand in Germany.

While self-driving cars with a multitude of sensors are already being field-tested, all of the former mentioned inventions use a very limited information base originating from the sensors of the vehicle itself. They are thus limited by the line-of-sight — be it visual, ultrasonic or radar.

The development of wireless communications allows to cross this line-of-sight barrier by exchanging messages between vehicles over various distances and around or past obstacles. Paired with high accuracy position information, obtained from systems like

Global Positioning System (GPS), the field of *inter-vehicular communication* (IVC)[1] is established. A new class of safety applications surpassing the line-of-sight limits using information gathered by *Vehicle-to-X* (V2X) communication alone or combining them with data obtained from sensors of the vehicle is possible. Examples of these V2X applications like *intersection collision warning*, *road hazard warning*, *construction site information system*, *weather hazard warning* and many more are already studied in *Field Operational Tests* (FOTs) (e. g. *Sichere Intelligente Mobilität Testfeld Deutschland* (sim$^{\mathrm{TD}}$) [6] and *DRIVE C2X* [7]).

While FOTs return invaluable knowledge, they are quite expensive, as many vehicles need to participate in preplanned tests with preplanned routes and timings for every participating vehicle. And still the gained results might not be applicable to the rest of the world, but are limited to the area the FOT was conducted in. Furthermore, small changes needed in the software due to bugs or optimizations result in huge costs as every vehicle's software needs to be updated and the tests need to be rerun.

1.1 Motivation and Problem Statement

Currently, two different technological approaches for IVC communication compete with each other. *Vehicle ad hoc Networks* (VANETs) using the *Institute of Electrical and Electronics Engineers* (IEEE) 802.11p *Wireless Local Area Network* (Wi-Fi) standard, specially designed for IVC communication (depicted in Figure 1.1), and the infrastructure-based mobile cellular network communication (shown in Figure 1.2). Both communication technologies offer quite different network characteristics like different available data rates, delays or drop rates. Similarly, each IVC application has its own communication characteristic needs and often it is not clear which communication technology best fits for a specific IVC communication application.

A common approach to reduce the need for time consuming, expensive FOTs are simulations. And while multiple simulators for IEEE 802.11p exist, the situation differs for cellular network simulations. The complexity of mobile cellular networks and their diversity, partly caused by their fast development process which leads to multiple cellular network generations working in parallel — make it hard to simulate every aspect of the network. Thus only few cellular network simulators exists (see Section 2.1), which are either highly probabilistic or based on confidential knowledge, which either is not available to the research community or only if a *Non-Disclosure Agreement* (NDA) is signed.

[1] C2X, V2X and IVC are synonyms for inter-vehicular communications. They are used interchangeably in the literature and in this thesis.

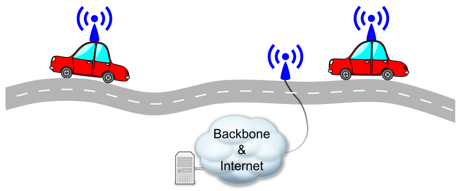

Figure 1.1: V2X communication using 802.11p Wi-Fi.

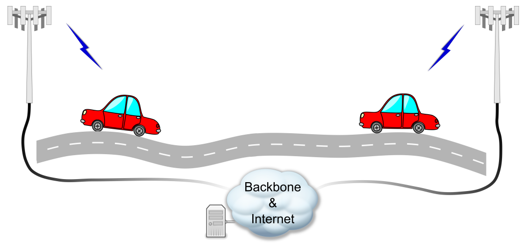

Figure 1.2: V2X communication using cellular networks.

We therefore propose to use a trace-based simulation approach to simulate mobile cellular network communication for V2X simulations. Instead of simulating every tiny aspect of the cellular network including, amongst others, signal propagation (which requires knowledge of e. g. obstacles, the exact antenna placements, sectorizations and frequency spectrum used), protocol handling, cell-handover and resource allocation and scheduling algorithms (which typically are influenced by the business plan of the ISP and thus confidential), we propose to treat the cellular network as a black box. Thus we aim to develop a trace-based simulation model for cellular network simulation that solely relies on cellular network measurements conductible by everyone and everywhere.

This approach raises numerous research questions in at least four fields of research: *Network measurement*, *map-matching*, *network simulation* and *coupling of multiple simulators*. As research is never straight forward, the order of the questions presented in the following paragraphs does not necessarily represent the timely ordering of the questions. This is especially true as we grouped them by research fields, but questions of one research area raise questions in other research areas.

For trace-based simulations, network measurements are extremely important. Thus even prior to the identification and evaluation of measurement methods multiple research questions in the field of network measurements arise:

- Which network characteristics can we measure? What does our measurement setup need to look like? Can we gather measurements for every geographical point in the simulation area or do we have to focus on parts of the map? How can we measure available data rates, loss rates and delays? How do we ensure synchronous clocks of the measurement devices to allow one-way delay measurements? Is it possible to simultaneously measure all network characteristics needed by our simulation model? If it is not possible to conduct simultaneous measurements of all needed network characteristics, is it possible and viable to combine multiple measurements of different characteristics? Can both communication directions be measured simultaneously? Regarding the measurement frequency: Which measurement frequency can we achieve and which measurement frequency does our simulation model need? How well does the measurement algorithm cope with fast changing network conditions of cellular networks? How good are the measurements we can gather, do they suffice for trace-based simulations and how do we prove this? How do we post-process the measurements? How do we cope with long phases of self-induced congestion in the post-processing and/or during the measurements?

On the one hand, mobile measurement devices usually log their position in geographical coordinates, namely latitude and longitude. These positions often are gathered by positioning systems like GPS, *Galileo*, *BEIDOU* or *GLONASS*, which all include measurement errors (compare Figure 1.3). On the other hand, traffic simulators typically simulate the movement of vehicles on coordinates derived from a road network-graph. Often they even use their own coordinate system for the simulation.

Transforming the error prone measurement positions onto coordinates compatible with a traffic simulator thus involves multiple steps, with map-matching at its first stage. Thus we are interested in the following research questions:

- Which map-matching algorithms exist and are they applicable to our measurements? How good do they cope with positioning errors? Can we alter the automatic mapping if we identify errors? Which mapping material can be used? Is it free of charge or do we need a license? As the timestamps of the measurements of the network characteristics most probably do not match those of the position measurements, how do we map-match the measurements? Which coordinate system does the used traffic simulator utilize? How can the map-matched positions be transformed into a format compatible with the traffic simulator?

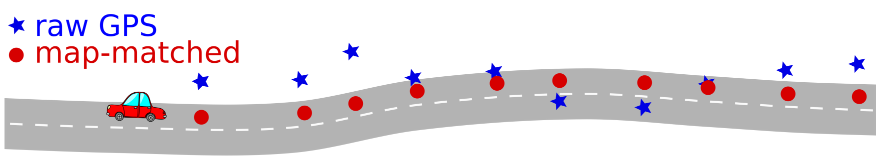

Figure 1.3: Map-Matching error prone GPS positions onto roads.

In the area of *network simulation* and in the area of *coupling of multiple simulators* we are most interested in the following research questions:

- What could our simulation model look like? How can it cope with the air interface and its possible higher loss rates than the cable-based backbone in the simulation? Which network characteristics need to be measured for our simulation model?

> Can we get the measurements our simulation model needs? How can we prove our simulation model works? How can we integrate our simulation model into a coupled simulation framework to allow everyone to do cellular network simulation for V2X application simulations. As our measurements are conducted using one mobile, cellular network connected, node and an Internet homed server we are also interested in the distribution of the measured available data rate on multiple competing mobile devices in the same network cell. Another interesting question is if it is possible to allow easy comparisons of a V2X application simulation using cellular networks to the same application simulation using IEEE 802.11p Wi-Fi.

Finally, after all these questions are answered, we aim to provide a complete simulation tool chain — from measurements via post-processing to trace-based simulation — for V2X applications using cellular networks.

1.2 Contributions

The rising mobility requirements of today's society lead to increased traffic densities, especially on the streets, which causes the number of accidents to increase. While modern driver assistance systems already help to reduce the number of accidents or at least to lessen their severeness, the vehicle's own sensors are limited to the line-of-sight. A promising approach to pass this barrier and to enhance the knowledge of each vehicle is to use V2X communication. By exchanging messages via IEEE 802.11p Wi-Fi or mobile cellular networks V2X applications can help to further enhance traffic safety and traffic efficiency. Different V2X applications are developed that serve different purposes, but also have different network requirements. Deciding which communication technology works best for a V2X application often is not trivial and FOTs are expensive. Simulations can vastly reduce development times and costs, but no open simulator for cellular networks exists that does not rely on confidential knowledge for its detailed simulation or which is not purely probabilistic.

The goal of this thesis is to evaluate the feasibility of trace-based simulation of cellular networks for V2X simulations, and furthermore to develop, evaluate and release a cellular network simulation model and couple it with a traffic simulator and a V2X application simulator. To achieve these goals we answer a multitude of questions stated in Section 1.1 in two papers, a journal article and a technical report presented in this thesis.

Our first contribution [1] presented in Chapter 3 shows the general feasibility of trace-based simulation of cellular networks by presenting our basic measurement method and the first draft of our simulation model. It analyzes published measurement methods for available data rate measurements and evaluates if they can cope with the fast changing network conditions featured by cellular networks. Furthermore it checks if and how available data rate measurements can be combined with one-way delay measurements. The paper thereby tackles some of the fundamental questions in the field of network measurements, but raises multiple others, like how to prevent long phases of *self-induced congestion* (SIC) when using *bulk traffic measurements*. While the paper does not solve this question, it offers post-processing workarounds reducing the negative impact of such SIC phases on the simulation results. By introducing and evaluating our simulation model the paper finally demonstrates the feasibility of trace-based simulation of cellular networks and along the way answers some research questions regarding the simulation model.

Our second contribution [2] (see Chapter 4) is entirely dedicated to answer the remaining research questions of the network measurement area unanswered by our first contribution [1]. By adding multiple feedback mechanisms and an *inter-chirp self-induced congestion* (ICSIC) prediction algorithm to our measurement methodology, we are able to largely reduce the length of ICSIC phases while at the same time raising the measurement accuracy and the measurement frequency. The resulting implementation of our measurement algorithm, the *Rate Measurement Framework* (RMF), is introduced, evaluated and released in this journal article [2]. The RMF allows the simultaneous, high frequency, position-based measurement of available data rates, delays, loss rates and modem status information for both communication directions between a mobile cellular network connected, moving node and a stationary, Internet homed server as depicted in Figure 1.4. To our best knowledge our measurement algorithm is the only available measurement algorithm that allows these simultaneous, high frequency measurements. By publishing its sources under MIT License at github we make the RMF available to everyone.

With our third contribution [3] presented in Chapter 5, we address the research questions arising in the map-matching area and regarding the preparation of the measurements to become the basis for our simulation model. The technical report introduces our complete post-processing tool chain and explains how we map-match our measurement traces onto a road network graph derived from *OpenStreetMap* (OSM) mapping data. Furthermore, it outlines how measurements of different traces can be combined without sacrificing realistic trace-based cellular network simulation. It also introduces multiple different filters allowing to prepare simulations with desired network characteristics that still mirror real world measurements. And finally, it explains why we chose a graph-based representation of our post-processed measurements as a basis for our simulation model. These explanations are accompanied by a walk-through

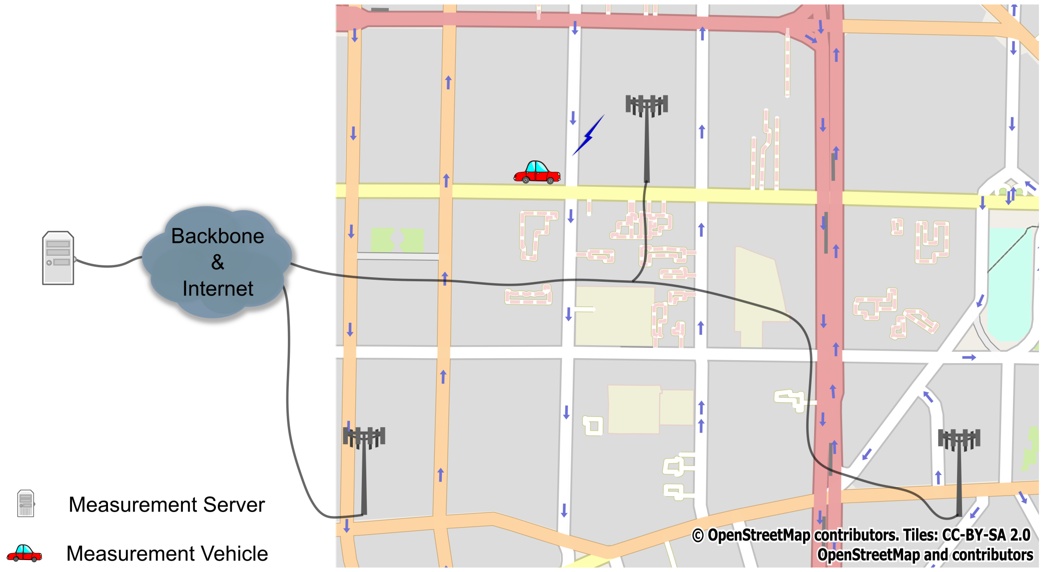

Figure 1.4: Setup for cellular network measurements (Map © OpenStreetMap [8]).

of our complete post-processing tool chain using example traces. The complete tool chain again is published as source code and made publicly available at github to allow anyone to post-process our or their own RMF traces.

The fourth and final contribution [4] of this thesis is presented in Chapter 6. It builds upon all previous contributions, enhances the simulation model presented in [1] by adding a cell share model, and highlights our researches by incorporating our simulation model in a fully-fledged coupled simulation environment for V2X applications, cellular network communication and road traffic. In this paper we present our enhanced simulation model as an *Objective Modular Network Testbed in C++* (OMNeT++) module which can be integrated in any network simulation based on OMNeT++. By utilizing the *V2X Simulation Runtime Infrastructure* (VSimRTI) framework, we are able to couple our OMNeT++-based cellular network simulation module with the traffic simulator *Simulation of Urban Mobility* (SUMO) and the V2X application simulator *V2X application simulator* (VSimRTI_App). This combination allows V2X application simulation with realistic cellular network communication simulation. Furthermore we show the usability of the presented simulation framework by implementing the *European Telecommunications Standards Institute* (ETSI) defined *Emergency Warning Application* (EWA) V2X application, which was defined for IEEE 802.11p Wi-Fi and our own, cellular network optimized, enhanced EWA version. With trace-based simulations based on measurements conducted with our RMF in an industrial area in Germany, which we post-processed with our map-matching tool chain, we are able to show the difficulties V2X applications defined for IEEE 802.11p face when used via cellular networks. With the simulation results of the standard EWA application we are able to develop an enhanced EWA version that, as demonstrated by further simulations, copes much better with the cellular network characteristics observed in the simulation area. Thereby the paper underlines the importance of trace-based cellular network simulation for V2X application development. As we furthermore published all sources of our simulation model and the simulation scenarios on our website, anyone can reproduce our results and create coupled V2X application, road traffic and cellular network simulations of their own utilizing our simulation model.

Thus, finally, a complete simulation tool chain for trace-based cellular network simulation for V2X simulations is available. By conducting measurements with the RMF in the desired simulation area and by using our post-processing tool chain, anyone can use our simulation framework to conduct realistic V2X application simulation using cellular networks.

1.3 Outline

This chapter explains the need for V2X application simulations and introduces the two competing network technologies used for V2X communication: IEEE 802.11p Wi-Fi and mobile cellular networks. It further outlines that a simulation gap for the simulation of mobile cellular networks exists. Furthermore it presents our contributions which lead to a complete simulation tool chain for V2X application simulation using cellular networks. This thesis is structured as follows:

Chapter 2 gives an extensive overview of the related work in the fields of cellular network simulation and coupling of multiple simulators for V2X simulations. It also explains our choice for OMNeT++ as the underlying framework for our cellular network simulation framework and the use of VSimRTI for the simulator coupling.

Chapter 3 presents our first contribution, which demonstrates the feasibility of trace-based cellular network simulation for V2X simulations. The paper presented in this chapter introduces our basic measurement algorithm based on *bulk traffic*, and our basic trace-based simulation model. It proves the feasibility of trace-based simulation by comparing real-world measurement traces with simulated measurements. Thereby this paper builds the fundament of this thesis. Based on the evaluation results presented in this paper we further improve our measurement methodology and our simulation model in our contributions presented in Chapters 4 and 6.

Our paper presented in Chapter 4 enhances the previous measurement method and introduces our Rate Measurement Framework *(RMF)* that allows high frequency measurements of available data rates, loss rates and delays simultaneously for both communication directions. While it is usable in wired networks, it is developed for cellular network measurements and thus is able to cope with the fast changing network conditions present in todays cellular networks.

In Chapter 5 we outline how we map-match the network measurements gathered with the previously introduced RMF. Using an example, we demonstrate our complete tool chain that map-matches, filters and aggregates measurements of one or multiple network traces and transforms them into a graph-based structure usable by our simulation framework.

Our enhanced simulation model for cellular networks, its implementation as an OMNeT++ module and its coupling with a traffic simulator and a V2X application simulator are introduced in Chapter 6. Based on the measurements presented in Chapter 4, which are post-processed with the tool chain published in Chapter 5, we again demonstrate the feasibility of trace-based cellular network simulation for V2X applications. Using the presented simulation framework, we implement an ETSI defined V2X application EWA and an enhanced, cellular network optimized version of EWA. On the one hand, this demonstrates the usability of our simulation environment and on the other hand, it emphasizes the invaluable use of simulations when porting V2X applications developed for IEEE 802.11p Wi-Fi to cellular networks.

All of our software developed and introduced in Chapters 4 to 6 is published on our website or github and is usable and modifiable by anyone. For details please consult the corresponding chapters and papers.

Chapter 7 concludes this thesis and discusses possible future work.

2
Related Work

The aim of this thesis is to provide a simulation framework allowing the realistic, reproducible simulation of *Vehicle-to-X* (V2X) applications and their cellular network communication without the need to have confidential knowledge of the cellular network setup deployed in the simulation area.

This goal is achieved by using a three step approach. First of all, key cellular network characteristics are measured using a mobile node traveling on all roads in all directions in the simulation area and measuring against a stationary Internet homed server. In a second step, the measurements are post-processed, map-matched and stored in a database. In the final step, this database is used to provide realistic, position-based cellular network characteristics to our trace-based simulation model. Coupling a road traffic simulator with a network simulator incorporating our cellular network simulation model and with a V2X application simulator allows realistic V2X simulations using cellular networks.

Our goal is partly achieved by studying and using prior work. This chapter thus gives a thorough overview on the related work for this thesis. As our paper *Moving measurements: Measuring network characteristics of mobile cellular networks on the move* [2], which is presented in Chapter 4, already includes a quite complete summary of related work on network measurements, the following sections focus on two large fields of related work. Section 2.1 focuses on related work in the field of cellular network simulation, Section 2.2 gives a thorough introduction into currently available simulation frameworks allowing V2X simulations by coupling simulators of different domains.

2.1 Simulation of Cellular Networks

Besides our trace-based cellular network simulation model, introduced in the papers presented in this thesis, other simulation models for cellular networks exist. Each of them has its benefits and drawbacks and differs in its possible use cases. This section summarizes prior and current work in this simulation domain and explains the differences to our simulation model.

Most cellular network simulators are developed focusing on the — at that time — current expansion stage of the cellular network technology, mainly with the intention to allow the evaluation of different optimization strategies in the cellular network while the hardware was not available. Thus different simulators for *Universal Mobile Telecommunications System* (UMTS), *High Speed Downlink Packet Access* (HSDPA), *Long Term Evolution* (LTE) and *Long Term Evolution Advanced* (LTE-Advanced) exist. Most researchers relied on already established network simulators as a basis for their simulation models. By this they did not have to implement every networking aspect, but could rely on well-tested code for common network protocols and could focus on the implementation of cellular network specific parts of the simulator. Hence the following subsections summarize the cellular network modules developed, grouped by the underlying network simulator.

2.1.1 Cellular Network Simulation based on Network Simulator (ns-2)

ns-2 [9] is a discrete event network simulator written in C++ and the *object-oriented extension to Tcl* (OTcl). Hence it runs on Linux or Windows using cygwin. Amongst others, it supports UDP, TCP, routing, multi cast protocols and wired and wireless networking (except cellular networks). The development on ns-2 has slowed down in favor of its completely rewritten "successor" *Network Simulator* (ns-3) (see Section 2.1.2). The current version ns-2.35 was released in November 2011. While ns-2 itself does not offer support for wireless cellular networks, some extensions were developed to add cellular network features.

Enhanced UMTS Radio Access Network Extensions for ns-2 (e.u.r.a.n.e) [10] adds support for the three UMTS node types *Radio Network Controller* (RNC), *Base Station* (BS) and *User Equipment* (UE). Furthermore it adds most of the UMTS and HSDPA protocol stack to ns-2. The mobile fading channels are simulated using MATLAB. E.u.r.a.n.e was developed to study scheduling effects of nearby and far away users. Hence the distance of each simulated UE is fixed during the simulation. Thus e.u.r.a.n.e lacks support for mobility, except for circular movement around the BS with a fixed radius as used in [11]. Furthermore the lack of multi cell simulations and handover mechanisms prohibit its use for V2X simulations. The project seems to be abandoned, as neither the project's website[1], nor the github project[2] hosting its sourcecode are available anymore. Thus we added a link to an ancient version of the project's website using the Internet Archive WaybackMachine to the reference [10].

The authors of [12] extend e.u.r.a.n.e to allow detailed HSDPA simulations. Their extended simulation model supports cell handover and user mobility using random way point mobility models. Additionally, the authors introduce extensions that allow to add new flow control algorithms to e.u.r.a.n.e. While these enhancements allow multi cell (multiple Base Stations) simulations, the simulations are still quite limited, as all simulated BSs are connected to a single RNC. Thus these simulations are not suitable for the purpose of the simulation of cellular network traffic between V2X applications.

2.1.2 Cellular Network Simulation based on ns-3 and LTE-EPC Network simulAtor (LENA)

Like ns-2, ns-3 is a discrete event network simulator which targets research and educational use. As such it is licensed under the GNU GPLv2 license. Ns-3 is written using C++and python. Thus it is not downward compatible to ns-2, which uses OTcl instead of python. Ns-3 is designed to support network virtualization and real testbed integration. It supports models for IP (v4 and v6) and non-IP based networks. The IP models focus on IEEE 802.11, WiMAX, 3GPP LTE and a variety of MANET routing protocols.

The development for **An LTE module for the ns-3 network simulator [13]** started during the *Google Summer of Code* (GSoC) 2010, where the ns-3 project was participating as a mentoring organization. The initial implementation focuses mainly on the *evolved UMTS Terrestrial Radio Access* (E-UTRA) part of the LTE system. It offers basic implementation of LTE devices and propagation modules for the *Physical layer* (PHY) and the *Medium Access Control* (MAC) layer. While it does not allow to simulate a complete LTE network, it already enables to simulate several important aspects of LTE systems like *Radio Resource Management* (RRM) and MAC scheduling. The authors [13] had to make a large number of assumptions to make the system work fast enough. For example, they did not simulate the flow of control messages and therefore every control message is always received correctly.

The code of the LTE module is included in ns-3 as of version ns-3.10, which was released in January 2011. Later on, the development of the LTE module was mainly driven by the *LTE-EPC Network simulAtor* (LENA) [14] project. This open source

[1] no longer available: `http://eurane.ti-wmc.nl/eurane/`
[2] no longer available: `https://github.com/metrue/ns2-umts`

project, which is based on ns-3, targets “the design and performance evaluation of Downlink & Uplink Schedulers, Radio Resource Management Algorithms, Inter-cell Interference Coordination solutions, Load Balancing and Mobility Management, Heterogeneous Network (HetNets) solutions, End-to-end QoE provisioning, Multi-RAT network solutions and Cognitive LTE systems” [14]. The enhancements to the LTE module made by the LENA project are frequently merged into the ns-3 LTE module.

The ns-3 LTE module currently (as of ns-3.26) is still limited to IPv4.

An import enhancement to ns-3 are the *Integrated Wireless and Traffic Platform for Real-Time Road Traffic Management Solutions* (iTETRIS) extensions (see Section 2.2.5), which add support for an *European Telecommunications Standards Institute* (ETSI) *Intelligent Transportation Systems* (ITS) compliant *inter-vehicular communication* (IVC) protocol stack (including for example *Cooperative Awareness Messages* (CAMs) and *Decentralized Environmental Notification Messages* (DENMs)).

In contrast to our trace-based simulation approach, the simulation of cellular networks with ns-3 currently is limited to LTE simulations only. Furthermore the design goal of the ns-3 LTE module was to enhance protocols, algorithms and schedulers in the cellular network. Using this simulation model for realistic V2X application simulations would require insights into the scheduling algorithms, antenna placement and many more aspects out of the *Non-Disclosure Agreement* (NDA) domain of a cellular network provider. And while LTE is rolled out by the cellular network providers, it still is not as widely available (especially in rural areas) as *General Packet Radio Service* (GPRS) and UMTS. While LTE is being pushed by the cellular network providers, it still is not as widely available as GPRS and UMTS networks. Thus V2X simulations using the ns-3 LTE module are limited to scenarios with LTE cellular network coverage only.

2.1.3 Cellular Network Simulation based on Objective Modular Network Testbed in C++ (OMNeT++)

OMNeT++ is a discrete event network simulation platform, which is free for non-commercial use. Its corresponding commercial version is named *OMNEST*. OMNeT++ comprises an *Integrated Development Environment* (IDE) (based on Eclipse), an execution environment and a simulation kernel. OMNeT++ modules are programmed in C++ and can be assembled into larger components by using a high-level *NEtwork Description* (NED) language. OMNeT++’s extensive GUI support allows to closely monitor and analyze the simulations graphically and textually.

The framework allows easy integration of third party libraries like MiXiM (IVC simulation), the INET framework [15], Oversim (p2p) and SimuLTE (LTE and LTE-Advanced). Further available modules include models for IEEE 802.11p, DSRC, IEEE 1609.4 WAVE, ETSI ITS G5 protocol stack and DCC.

Cellular network simulations in OMNeT++ can be conducted as described in **Simulating LTE/LTE-Advanced Networks with SimuLTE** [16]. *LTE User Plane Simulation Model for INET & OMNeT++* (SimuLTE) uses the modular programming practices of OMNeT++ and allows simulations of LTE and LTE-Advanced. By using other OMNeT++ modules, it is also possible to simulate multi communication platform scenarios (*Wireless Local Area Network* (Wi-Fi) and LTE).

SimuLTE is released under the LGPL license and makes use of well-tested protocol stacks shipped with OMNeT++ and the INET framework. It simulates the data plane of the LTE *Radio Access Network* (RAN) and Evolved Packet Core in *Frequency Devision Duplex* (FDD) mode with heterogeneous eNBs (macro, micro, pico etc). It features realistic channel models, MAC and resource scheduling for both communication directions. Additionally, it supports the simulation of omni-directional and anisotropic antennas and includes HARQ functionalities. The authors of [16] state that currently *Radio Resource Control* (RRC) is not modeled.

The main use of SimuLTE is the study of the influences of resource scheduling. Simulating V2X scenarios with SimuLTE would require deep knowledge of the simulated cellular network - which normally is not available and under NDA.

In **Modeling unicast device-to-device communications with SimuLTE** [17], the authors enhance SimuLTE by adding direct *device-to-device communication* (D2D) between LTE devices residing in the same network cell. D2D will probably be used in future generations of V2X communication devices, as it allows direct communication for vehicles residing in the same network cell. Thus the two-way traversal of unicast packets over the *Evolved Node B* (eNB) it not necessary anymore.

D2D uses part of the cells' uplink frequency resources where less severe interference is expected. A UE signals new D2D backlog using the *Random Access Channel* (RAC) and the eNB then assigns resources for the D2D communication to the UEs.

While D2D might help to reduce the latency for inter cell V2X communications in the future, its use is limited to communication distances of the cell diameter. Thus it can not be used for a large number of V2X applications that require longer communication ranges. Furthermore, all the UEs and eNBs involved must support the new technology.

Compared to our trace-based simulation approach, SimuLTE with the D2D enhancements has a simulation advantage for inter cell communication in LTE cells. But it still needs NDA knowledge of the used cellular network and is limited to LTE and LTE-Advanced only.

2.1.4 Simulating Cellular Networks by modeling WCDMA and HSDPA downlink traffic

In their paper **HSDPA Performance in a Mixed Traffic Network** [18], the authors analyze the best way to split the transmission power between Release 99 *Wideband Code Division Multiple Access* (WCDMA) downlink channels and HSDPA's *High Speed Downlink Shared Channels* (HS-DSCHs) in a network cell. Therefore the authors implement their own simulation model for the key characteristics of both WCDMA and HSDPA downlink traffic. While their modeling of the WCDMA traffic is somewhat scarce, the HSDPA traffic is modeled more accurately. The authors' aim is to help decide at which ratio WCDMA and HSDPA nodes should be served in mixed mode cells sharing the same transmitter. The authors state that individual link level simulations for each user and base-station would result in unfeasible computation complexity. Thus they modeled the individual links through a simplified system level description.

As the presented simulation model was developed with the goal to decide how to split transmitting power and resources between WCDMA and HSDPA downlink channels, no network protocols are simulated. Combined with the focus on downlink traffic simulations only, their simulator is not suitable for V2X application simulations.

2.1.5 Simulating the LTE PHY layer

Simulating the Long Term Evolution physical layer [19] focuses on the simulation of the downstream PHY layer of LTE. The authors complain that (in 2009) only commercial LTE simulators exist, while they ideally should be publicly available to facilitate the comparison of work made by other researchers. Thus the missing open source LTE simulator impedes the work of researchers, which would like to study and enhance algorithms not completely specified in the LTE standard. Consequently, the authors publish a standard compliant, open source LTE downlink simulator, which utilizes a combination of MatLab and C functions for the simulation. Their simulator allows to simulate single-downlink, single-cell multi-user and multi-cell multi-user simulations. It includes *Adaptive Modulation and Coding* (AMC), *Multiple Input and Multiple Output* (MIMO) transmissions, *Hybrid Automatic Repeat Request* (HARQ) and *Channel Quality Indicator* (CQI) calculations, multiple users and scheduling.

The authors explain that the simulation of every aspect of the PHY layer is really complex. Thus they exploit multi-threading with multiple CPU cores utilizing Mathworks' Parallel Computing Toolbox [20].

Their simulator focuses on the downlink channel only and thus features an error-free uplink feedback channel with adjustable delay. Thus it is not suitable for V2X simulations - even when focusing on LTE only scenarios. Furthermore, as its purpose is to study LTE algorithms, it lacks support for common network protocols, which would need to be implemented to allow proper V2X communications.

2.1.6 Cellular Network Simulation with LTE-Sim

In 2010, the authors of **Simulating LTE Cellular Systems: an Open Source Framework** [21] discovered the need for an open source LTE simulator. As researchers, they disliked the need to purchase a commercial license for any of the available closed source simulators to study and evaluate the performance of the LTE system. Thus they developed their own event-driven LTE simulator in object-oriented C++, named it *LTE-Sim* and released it under the GPLv3 license.

Their intention was to allow to study and evaluate the performance of the entire LTE system with their simulator. *LTE-Sim* thus simulates upload and download scheduling strategies in multi-cell and multi-user environments. Therefore, the authors implemented UE, eNB, *Mobility Management Entity* (MME) and *Gateway* (GW) network nodes. Furthermore, *LTE-Sim* supports mobility, radio resource optimization, frequency reuse techniques, CQI calculation, CQI feedback and AMC.

The authors stress that a complete PHY layer simulation as performed in [18] or [19] (Sections 2.1.4 and 2.1.5) requires high computational effort and thus is not suited for complex network scenarios. They outline that the simulator presented in [19], which implements the complete LTE PHY layer, requires more than 3 hours to simulate few seconds of a network composed of only 3 nodes. Thus the authors decide to use an analytical model instead, which significantly reduces the computational complexity of the simulation.

While LTE-Sim does not rely on as much NDA knowledge of the simulated cellular network details, its use for V2X simulation is still limited to LTE only scenarios. Furthermore LTE-Sim is not based on a network simulation framework and thus many networking protocols would need to be implemented to offer a complete V2X network stack.

2.1.7 VSimRTI Cellular Simulator

The *V2X Simulation Runtime Infrastructure* (VSimRTI) framework ships with its own cellular network simulator that consists of a GEO Server and the *Cellular Network Simulator* (CNS). The GEO Server keeps track of the locations of all vehicles and is used to incorporate Geo addressing. The CNS is responsible for simulating the packets' traversal through the cellular network. It divides the simulation area into multiple independent polygonal regions that feature their own network characteristics. Three different modules are responsible for the simulation of the packet traversal through the cellular network. The *Core Delay Model* simulates the delay of every packet and supports multiple basic delay types. The packet losses between the mobile node and the base station are simulated by the *Packet Delivery Ratio Model.* And finally, the *Bandwidth Model* calculates the delay for individual packets. taking the channel load of the vehicles' region into account.

The simulation model used in *VSimRTI Cellular Network Simulator* (VSimRTI_Cell) features some similarities to our trace-based simulation model as it also treats the cellular network as a black box and uses definitions for the same network characteristics, delay, loss rate and available data rate. In [22] the authors of VSimRTI_Cell state that they use measurements from different campaigns to develop their simulation models. On the one hand, they suggest that their free region definition could be used to simulate cellular networks based on position-based network characteristics gathered through measurement campaigns or crowd-sourced data. On the other hand, no mapping to base stations is performed. In comparison, our simulation tool chain presented in this thesis post-processes the measurement traces and identifies the network cell involved in the transmission of every measurement packet. Our cell share model uses this information to distribute the available data rate on a cell based level. Furthermore, in contrast to VSimRTI_Cell, our simulation model considers the direction of travel. This is quite important, as cell handover is conducted partly based on signal strength comparison of neighboring cells. Thus cell handover typically does not happen at the same position on the road when traveling in different directions.

2.1.8 Conclusion on current Cellular Network Simulators

The related work on Cellular Network Simulators presented in the previous sections features a wide variety of simulation approaches and simulation goals. Most cellular network simulators are developed to study and optimize the performance of an evolution of a cellular network (like HSDPA, LTE or LTE-Advanced). This often makes them accurate for their targeted simulation purpose, but leaves large open spots for other simulation needs. Thus it is hard to find a simulator capable of simulating a wide range of cellular network technologies. And while LTE is spreading fast it still will take some time until every spot of the world has LTE coverage, thus a simulator for V2X simulations using cellular networks should incorporate at least GPRS, UMTS, *High Speed Packet Access* (HSPA) and probably LTE. Furthermore, it is desirable that such cellular network simulators base on well-established and tested network simulation frameworks like ns-2, ns-3 or OMNeT++. Their large communities, many extensions and supported network protocols help to bring together all needed simulation aspects for complete V2X cellular network communication simulations.

Additionally, the presented papers show that the simulation of all aspects of cellular networks (like a complete PHY layer simulation) is computationally quite expensive and not suitable for large scale simulations.

Our cellular network simulation model, introduced in chapter 3, offers a low computational complexity by utilizing cellular network traces as its simulation basis. By enhancing and publishing it as an open source OMNeT++ module (presented in chapter 6), it can additionally be used in almost any OMNeT++ simulation and make use of the many well-established OMNeT++ features and third party modules.

Furthermore, our simulation model does not rely on any confidential or NDA knowledge of the simulated cellular network, as it only relies on cellular network traces, which can be conducted using standard x86 hardware and our open source measurement framework presented in chapter 5.

2.2 Coupling multiple simulators for V2X simulations

Realistic simulations of V2X applications require the combination of simulators originating in different fields of knowledge. While it might be possible to create a simulation environment incorporating all aspects of road traffic simulation, network simulation and V2X application simulation in a single simulator, this approach would have many disadvantages. It would be hard to gather the expertise in all fields of knowledge and it would also consume a significant amount of time for planning, programming and verification of the combined simulator. Additionally, every simulation aspect would have to be verified by experts to gain reputation in the science community and thus gain trusted simulation results.

Hence the approach to couple well-established simulators of the different domains is much more promising. At least three simulators need to be coupled to allow realistic simulations of V2X applications communicating via cellular networks. First of all, it is vital to properly simulate the flow of the network traffic of all equipped vehicles. Thus it is important to incorporate a network simulator with cellular network simulation capabilities. Additionally, a well-established road traffic simulator is needed to properly simulate the traversal of vehicles on the road network. And finally, a simulator for the V2X application itself must be included.

To allow the accurate simulation of as many thinkable V2X applications as possible, those simulators have to run simultaneously and must allow interactions between the simulators. The V2X application for example might need to be able to change the route of the vehicle, and the cellular network simulation might need to react to position changes of the vehicle. The next section gives a brief summary of a book chapter on V2X simulations. Further sections, which we use to give an overview on well-known and established V2X simulation frameworks, follow.

2.2.1 Simulation Tools and Techniques for Vehicular Communications and Applications

The book chapter **Simulation Tools and Techniques for Vehicular Communications and Applications** [23] is an excellent summary of current V2X application simulation. The authors of this book chapter are actively involved in the development of the three major simulation frameworks for coupled V2X-application simulation: *Vehicles in Network Simulation* (Veins), iTETRIS and VSimRTI. They give an extensive overview on simulation models specific to the *inter-vehicular communication* (IVC) domain. Furthermore, they describe the required methodology and explain how to choose a suitable simulation approach for the desired granularity and scalability of the simulation and the simulation results. This is achieved by analyzing and categorizing IVC Applications, Wireless Communications & Networking and Mobility Modeling. Additionally, the authors give an overview on major network and mobility simulation tools, namely ns-3, OMNeT++, *Java in Simulation Time / Scalable Wireless Ad hoc Network Simulator* (JiST/SWANS), *Simulation of Urban Mobility* (SUMO) and *Verkehr In Städten - SIMulationsmodell* (VISSIM). Finally, they briefly describe the benefits and drawbacks of the simulation frameworks Veins, iTETRIS and VSimRTI. They conclude the book chapter [23] by stating:"In general, the credibility of IVC simulation studies, and most importantly, the reproducibility can substantially be increased by using just one of these toolkits."

2.2.2 Mixed Simulator (MiXiM)

The simulation framework *Mixed Simulator* (MiXiM) [24], [25] is based on the OMNeT++ simulation environment. It supports the modeling and simulation of wireless networks and includes environment models, connectivity and mobility models, handles reception and collision simulation and includes a protocol library. While it simulates node mobility using the *Mobility Framework* [26]. it does not offer an interface to other simulators like a road traffic simulator. Thus it can not directly be used for coupled simulation of road traffic, cellular network and V2X applications. Furthermore MiXiM is deprecated and the authors suggest to use the OMNeT++ module *INET* 3.x instead.

2.2.3 Vehicles in Network Simulation (Veins)

Veins [27] combines the network simulator OMNeT++ with MiXiM and the road traffic simulator SUMO. For simulation control and data collection, the OMNeT++ simulation kernel is used. By providing a bidirectional coupling using the *Traffic Control Interface* (TraCI) [28], Veins allows feedback and reactions on simulated events. The TraCI control protocol offers an interface to all available SUMO values and allows changes of some of these values. Veins is a modular framework, which allows to simulate V2X applications. It offers no standardized protocol for inter-simulator communication and thus is not easily extendable to include other simulators. As it uses MiXiM, Veins does not offer cellular network simulation. Source code, tutorials and related publications are freely available on the Veins website [29].

2.2.4 Vehicles in Network Simulation merged with SimuLTE (VeinsLTE)

VeinsLTE [30] is a simulator for heterogeneous vehicular networks based on Veins. As it derives from Veins, it still uses the road traffic simulator SUMO and OMNeT++ with the MiXiM modules for *Institute of Electrical and Electronics Engineers* (IEEE) 802.11p simulation. Additionally, it incorporates SimuLTE (see Section 2.1.3) for detailed LTE simulations. As SimuLTE was developed without the idea of LTE nodes entering and leaving the simulation area during the simulation, the authors had to add this feature to the SimuLTE LTE stack. The resulting VeinsLTE offers the V2X application the possibility to either directly choose which network stack to use (IEEE 802.11p or LTE) or to delegate this decision to a decision maker module. VeinsLTE is available for download on the VeinsLTE website [31].

2.2.5 Integrated Wireless and Traffic Platform for Real-Time Road Traffic Management Solutions (iTETRIS)

The EU-funded open source simulation platform iTETRIS [32], [33] allows to study large scale cooperative Intelligent Transportation Systems. By extending ns-3 and SUMO, iTETRIS allows the coupled simulation of road traffic, network traffic and the simulation of V2X applications. The authors introduce the *iTETRIS Control System* (iCS), which is responsible for the interaction of SUMO and ns-3. V2X applications are simulated as *iTETRIS Applications* (iAPPs). The iCS subsequently triggers the simulators ns-3, SUMO and iAPP to execute their tasks for the currently simulated second. The default simulation granularity of one second was implemented with large scale simulations in mind, however the authors of [33] state that the granularity can be modified to achieve smaller time steps. The network simulation is conducted with ns-3 and follows the ETSI's ITSC architecture. It supports the access technologies ITS G5A (an evolution of the IEEE 802.11a standard), *WiMAX*, UMTS and *DVB-H*. The iTETRIS authors port an ns-2 UMTS simulation module [34] to ns-3 and simplify those parts that relate to the management of UMTS networks, to allow UMTS simulations in iTETRIS. Furthermore, the authors add support for an ETSI ITS compliant IVC protocol stack to ns-3.

2.2.6 V2X Simulation Runtime Infrastructure (VSimRTI)

VSimRTI [35] is a simulation runtime infrastructure that is able to couple multiple simulators and thereby offers unified simulation of different V2X features. The VSimRTI core is bidirectionally coupled with each simulator using an *ambassador* and a *federate* per simulator. The *ambassador* of a simulator defines which messages the simulator is interested in and which messages it can provide. The VSimRTI core forwards all incoming messages to the ambassadors of all simulators that are interested in this message type. An ambassador receiving a message hands it to its corresponding *federate*, which translates the message into the "language" of the simulator. This interface structure allows to easily add new simulators to VSimRTI. Furthermore, this design makes it possible to quickly change the simulation setup by switching simulators, like e. g. the network simulator. VSimRTI already offers a fully-fledged V2X application simulator and its own cellular communication simulator, VSimRTI_Cell (see Section 2.1.7). Additionally, the authors implemented *ambassadors* and *federates* for multiple simulators like SUMO, ns-3, OMNeT++ and JiST/SWANS (the last one was removed in Version 0.17). VSimRTI is available for download on the authors' web pages [36]. It is partly closed source, but the majority of the simulators are open source.

2.2.7 Network Simulation and Emulation Platform (NetSim)

Network Simulation and Emulation Platform (NetSim) [37] is a commercial network simulation and emulation platform. Regarding its advertisement [38], it offers a wide variety of simulation, emulation and analyzing possibilities. Amongst others, it offers the simulation of cellular networks (GSM, CDMA, LTE, LTE-Advanced) using special components (requiring extra licenses). Furthermore, it allows the combination of network emulation and real hardware running live applications. Thus it offers some of the features needed for V2X application simulation using cellular networks. Unfortunately the SUMO interface is only available for *VANETS* and thus a coupled simulation of traffic and cellular network is not supported.

2.2.8 Conclusion

After a thorough analysis of the related work on coupled V2X application simulation, we chose to use VSimRTI for our simulations for multiple reasons. First of all, VSimRTI already ships with the broadest range of supported simulators. Furthermore, the *ambassador* and *federate* interface structure allows to easily integrate currently unsupported simulators. Most important, it perfectly fits our simulation needs as it already offers a V2X application simulator and integrates SUMO and OMNeT++. Therefore only minimal changes are needed to integrate our simulation model. And while part of the source code of the VSimRTI core is closed source, the maintainers offer great support and either incorporate needed changes or new features themselves, or, in our case, granted access to parts of the sources to incorporate the needed feature ourselves. Finally, VSimRTI is free for academic use and is still actively developed and maintained by a motivated team offering great support whenever questions arise.

3 Trace-based Simulation of C2X-Communication using Cellular Networks

This chapter summarizes the contributions of the paper [1]:

> Norbert Goebel, Markus Koegel, Martin Mauve, Kalman Graffi:
> *Trace-based Simulation of C2X-Communication using Cellular Networks*
> In Proceedings of the 11th Annual Conference on Wireless On-demand Network Systems and Services (WONS 2014) - Special Session on VANETs and ITS, Obergurgl, Austria, April 2014.

Currently *Car-to-X* (C2X)[1]-Communications feature two competing communication technologies. Local, ad hoc short-range communication (e.g. IEEE 802.11p Wi-Fi) on the one hand, and conventional cellular network communication (e.g. GPRS, UMTS or LTE) on the other hand.

For the automotive industry, it is important to determine the impact of the used communication technology on the usability of any C2X application. A key factor of this impact is the availability of information in the networked vehicles. Often simulations are used during the development process to reduce the need for expensive *Field Operational Tests* (FOTs). Additionally, simulations can help to significantly speed up the development process. In contrast to FOTs, they offer repeatable experiment setups and thus allow to test different implementation versions of the same C2X application with the same simulated network conditions.

While multiple well-established simulation models and simulators for 802.11p Wi-Fi exist, it is much harder to simulate mobile cellular networks due to their complex

[1]C2X, V2X and IVC are synonyms for inter vehicular communications used interchangeably.

nature. Thus no well-established and evaluated simulators for cellular networks exist that do not rely on confidential material owned by the network operators, which is not available to the public.

The goal of our paper [1] thus is to evaluate the feasibility of simulating a cellular network for C2X-communications based on traces of the cellular network.

3.1 Paper Summary

This paper outlines how the simulation gap in C2X-simulations using cellular networks can be filled by trace-based cellular network simulations.

Its key contributions are:

1. The paper explains how cellular network traces can be conducted using our measurement methodology to obtain simultaneous measurements for available data rates, delays and loss rates for both communication directions of a mobile node to/from an Internet homed server.

2. We introduce and validate our trace-based simulation model for cellular networks for C2X-communication. As our proposed simulation model uses network traces as its simulation basis, it does not need any low level information on the cellular network. Instead, it treats it as a black box and completely relies on measurable network characteristics anyone can gather with standard, affordable x86 hardware.

To outline the novelty of our measurement methodology, the paper first gives a short review and evaluation of state of the art available data rate measurement methods and introduces the used nomenclature. It shows that neither of those available data rate measurement methods allows the simultaneous measurement of delays and available data rates. Furthermore we reveal that neither of them is designed for fast changing network characteristics cellular networks facilitate. Since our simulation model relies on measurements of available data rates, delays and loss rates for both communication directions, the paper introduces our own bulk traffic measurement methodology. Our measurements are conducted between a moving mobile node connected to a cellular network and an Internet homed server. We simultaneously measure network characteristics of both communication directions. During the measurements, the mobile node — which typically is vehicle-mounted — traverses all roads of the road network of the desired simulation area. As our measurements also log GPS positions of the mobile node, the gathered traces contain detailed, position-based network characteristics (mainly available data rates, delays, loss rates, modem status) for upstream and downstream of the cellular network. An exemplary simulation area is shown by the green roads in Figure 1 of our paper [1].

The basic concept of our measurement algorithm is based on packet trains — a sequence of packets sent back-to-back — that are sent in both communication directions simultaneously. This bulk traffic measurement leads to self-induced congestion — network buffers en route of our measurement packets are still filled up with packets of previous packet trains. While self-induced congestion does not harm available data rate measurements, it falsifies the one-way delay measurements conducted simultaneously with the first packet of each packet train. Hence our algorithm aims to divide each measurement second into two equal parts — 500 ms data packets transmitted followed by 500 ms of silence. During the silence period, the network buffers en route of the network path can empty, and thus the first packet of the next packet train can be used to measure the one-way delay of the network path. While the remaining packets (2..n) of the train are used to estimate the available data rate, all packets (1..n) of the packet train are used to calculate the packet loss ratio. As mentioned earlier, our measurement algorithm simultaneously conducts measurements for both communication directions. Thus it can use the payload of the measurement packets of one direction to transport the most recently measured network characteristics of the other communication direction. Using this piggybacked information on the cellular network characteristics, the number of packets and the size of the measurement packets are adapted in an effort to achieve the 500 ms packets transmitted and 500 ms silence periods per second.

Next, the paper introduces our basic simulation model, which is based on the assumptions that the wireless link is the tight link and that the vast majority of the packet drops occur on the wireless link. Both communication directions are modeled by two FIFO queues each. One queue of each direction is used to simulate the available data rate and loss rate and thus the air delay of the packets. The other queue of each communication direction simulates the backbone delay (see Figure 3 of our paper [1]).

After the description of the implementation details of our simulation model, our basic simulation model is evaluated by simulating the measurement traffic itself. By sending the measurement packets (logged in the network traces) at the exact same time in the simulation as during the real world measurements, we can monitor the simulated packets received.

We focused on two metrics comparing simulation and real-world traces during the evaluation: The differences in the end-to-end delay for each packet and the number of dropped packets. The evaluation shows that the basic simulation model works well in most situations, but faces problems in situations where the measurement algorithm was not able to impede self-induced congestion.

Based on these findings, we present five preprocessing enhancements to reduce the negative impact these long self-induced congestion phases impose on the simulation results. The evaluation of these enhancements shows that, while they significantly reduce the negative impact on the simulation results, the preprocessing can not completely eliminate the self-induced congestion phases. Regarding the simulation of packet drops, now only a ratio of 0.0000025 of all upstream packets and 0.0018 of all downstream packets are not dropped correctly. Furthermore, 87% of the downstream packet delays and 82% of the upstream packet delays are off by at most 3 ms in the simulation.

While these simulation results show that trace-based simulation of cellular networks is feasible, two challenging situations are revealed by the evaluation:

- long phases of self-induced congestion with high delays
- phases with 100% packet loss

Thus we conclude that further improvements to the measurement framework are needed to reduce the length of self-induced congestion phases.

3.2 Contribution

The first key contribution of this paper is the introduction of our measurement methodology, which allows to simultaneously gather position-based measurements for the available data rate, loss rate and delay for both communication directions of a cellular network. For that matter, the paper also outlines that no other available data rate measurement algorithm currently can conduct such simultaneous measurements.

The second key contribution of this paper is the introduction and evaluation of our trace-based simulation model for cellular networks for C2X-communication. In contrast to simulators aiming at simulating every single aspect of a cellular network, our simulation model does not rely on confidential, low level information on the cellular network. Instead, it treats the cellular network as a black box and completely relies on measured network characteristics anyone can gather with standard, affordable x86 hardware.

3.3 Personal Contribution

The reviewing of the selection process of IEEE/IFIP Annual Conference on Wireless On-demand Network Systems and Services (WONS) 2014 resulted in an acceptance ratio of 22.9%.

The author of this thesis, who is also the main author of the paper, Norbert Goebel, developed the measurement methodology and gathered the network traces presented in this paper. He also stated and implemented the simulation model — inspired by trace-based simulation of a predecessor of today's Wi-Fi. Furthermore, he conducted the simulations and engineered the enhancements presented in this paper. Additionally, he performed the analysis of the simulation results and discussed them with Markus Koegel, Martin Mauve and Kalman Graffi and wrote the paper with their editorial help.

3.4 Importance and Impact on the Thesis

The paper summarized in this chapter builds the foundation of this thesis. First of all, it outlines the cellular network simulation gap in current C2X simulations. Second, it introduces our basic measurement methodology to obtain sound cellular network traces. Furthermore, the paper introduces our simulation model, which uses these traces to accurately simulate packet traversals in the cellular network. And finally, we validate our simulation model by using a prototype implementation and identify possible areas for future work.

Based on the evaluation results of this paper, we greatly improved our measurement methodology, which we present and evaluate in chapter 4. Furthermore, we further enhance the simulation model presented in this chapter in multiple ways, implement it as an OMNeT++ module in chapter 6 and incorporate it into a fully-fledged coupled simulation framework for C2X simulations. And finally, the paper presented in chapter 5 shows how the enhanced network traces (chapter 4) can be processed to form the simulation basis for the cellular network and road traffic simulators of the coupled simulation framework (chapter 6).

4 Moving Measurements: Measuring Network Characteristics of Mobile Cellular Networks on the Move

This chapter summarizes the contributions of the paper [2]:

> Norbert Goebel, Tobias Krauthoff, Kalman Graffi, Martin Mauve:
> *Moving measurements: Measuring network characteristics of mobile cellular networks on the move*
> Published Journal Paper in Elsevier: Computer Communications, Volume 97, 2017.

Sound network traces are fundamental for trace-based simulations, but tracing cellular networks impedes tough challenges.

As outlined in our previous paper [1] (see Chapter 3), none of the established available data rate measurement tools (often named available band width measurement tools) is capable of simultaneous available data rate, drop rate and delay measurements. Furthermore these established measurement algorithms rely on assumptions not feasible in cellular networks. For example constant network conditions during a measurement period of multiple seconds are unlikely in cellular networks — especially with moving measurement devices tracing the network on all streets of a road network graph.

In [1] we introduced our bulk traffic measurement method allowing the aforementioned simultaneous measurements. In the same paper we outline that further improvements to this measurement methodology are needed to reduce the length of self-induced congestion phases and thus enhance the quality of the measurements in challenging network conditions.

The main goal of this paper [2] is to introduce and evaluate our novel measurement method that allows high frequency measurements of available data rates, drop rates and delays of a cellular network for both communication directions simultaneously. To reach this goal, phases of self-induced congestion have to be eliminated or at least be shortened as far as possible.

4.1 Paper Summary

This paper introduces and evaluates our novel measurement method that is capable of conducting high frequency, position-based measurements of the key network characteristics of mobile cellular networks while moving. It allows to simultaneously measure the application-to-application available data rates, delays, loss rates, *Global Positioning System* (GPS) positions and cellular modem information for the upstream and downstream direction from a moving mobile cellular network client to an Internet homed server. The paper tackles the measurement issues identified in [1] and identifies and solves further measurement challenges implied by the fast changing network characteristics of mobile cellular networks.

As sound network traces are fundamental for trace-based simulations, this paper addresses a key challenge for trace-based simulations of *Vehicle-to-X* (V2X) communication using cellular networks. The requirement for frequent measurements combined with the need for simultaneous delay and available data rate measurements in rapidly changing network conditions pose tough challenges addressed in this paper.

The key contributions of this paper are:

1. The introduction of our measurement method allowing frequent measurements of application-to-application available data rates, loss rates and delays simultaneously for both communication directions. Our proposed measurement method is able to adapt to fast changing network conditions of mobile cellular networks and reduces the length of *inter-chirp self-induced congestion* (ICSIC).

2. The evaluation of the implementation of our measurement method proving that the identified challenges induced by the high measurement frequency, the combined delay and available data rate measurements and the rapidly changing network conditions are solved.

The paper thereby also solves the challenges identified in our previous paper [1].

The paper first analyzes relevant, existing measurement methods for application-to-application available data rate measurements. These methods can be grouped into three categories: *Probe Gap Model* (PGM), *Probe Rate Model* (PRM) or *Bulk Traffic*.

We explain why all reviewed measurement tools (Spruce [39], IGI [40], Delphi [41], TOPP [42], Pathload [43], Pathchirp [44], Assolo [45], iperf2 [46], iperf3 [47] and ookla speedtest.net [48], [49]) are not capable of solving the challenge of frequent, simultaneous delay and available data rate measurements in rapidly changing network conditions. The paper further outlines that the best basis to solve this challenge is to use a heavily modified *bulk traffic* measurement method.

Based on this background knowledge, the paper introduces our basic measurement methodology, which fits into the *bulk traffic* category. It uses chirps for its measurements which each consist of packets sent back-to-back.

Bulk traffic measurements typically (as long as the sending rate is higher than the receiving rate) cause the build up of packet queues in the network path, which is called *self-induced congestion* (SIC). In contrast to ordinary bulk traffic measurements, we introduce gaps of silence between consecutive chirps — the so called *inter-chirp gap*. Thereby the first packet of each chirp hits network buffers that are not filled with previous measurement packets. At least if the length of the silence period is long enough. Therefore, we alter the chirp characteristics (packet size and number of packets per chirp) of each chirp based on feedback received from the opposing measurement node in an effort to reach a fixed utilization period and silence period in each measurement interval. If the inter-chirp gap is large enough and the clocks of both measurement devices are synchronized, the first packet of each chirp can be used to measure the one-way application-to-application delay. For this purpose, the paper explains how sub millisecond time synchronization between both measurement devices is possible by using pulse per second enabled GPS hardware.

Compared to the complexity to ensure empty network buffers prior to delay measurements, the available data rate and loss rate measurements are straight forward. The available data rate can be estimated by calculating the received bytes per second. The drop rate can be measured by calculating the ratio of received packets versus those sent. The paper explains the used nomenclature and formulas in detail.

It also identifies two major challenges for the needed high frequency measurements induced by the fast changing network conditions of cellular networks:

1. It is vital but not trivial to determine the correct length of each chirp and the correct length for the inter-chirp gap to avoid ICSIC.
2. Chirps can not be arbitrarily long as network buffers tend to overflow if too many packets arrive. This leads to massive tail drops — dropped packets at the tail of chirps caused by overflowing network buffers.

We explain why establishing a feedback loop by piggybacking recent measurements in the payload of the measurement packets works well for most situations, but does not allow to adapt to fast decreasing available data rates, which can lead to long phases of ICSIC. In [1] we observed ICSIC phases of up to 50 seconds. These ICSIC phases render delay measurements invalid and lead to reduced measurement frequencies for available data rates and drop rates. Thus we develop and introduce an ICSIC detection algorithm, which, upon reception of any packet, calculates an ICSIC free sending time for the first packet of the following chirp and piggybacks this information in the measurement payload, too. On reception of this information, the other measurement device can alter the transmit time of the next chirp accordingly and thus avoid ICSIC. Furthermore we describe how we filter delay measurements falsified by ICSIC and introduce a second physical cellular network link to reduce feedback delays.

Finally, we propose three independent methods we combined to solve the tail drop challenge. We throttle the sending rate using a self-written kernel module to overcome packet losses in our devices. Furthermore we throttle the sending rate to the technology limit of the modem and we change the length of each measurement interval to 100 ms chirps followed by 100 ms silence.

Prior to the evaluation part of the paper, we outline implementation details of the *Rate Measurement Framework* (RMF), which incorporates the previously described measurement methodology. Furthermore, we describe the various hardware setups used for our cellular network measurements presented in the paper.

The evaluation section of the paper focuses on four topics. First of all, we thoroughly analyze statistics of 9 network traces (see Table 3 of our paper [2]). Two traces were gathered with the predecessors of the RMF used for the measurements in [1]. Two were conducted using an older version of the RMF and five with the latest RMF version. The analysis clearly shows that the improvements introduced in the paper at hand solve the challenges identified and significantly improve the quality of the traces. All traces conducted with the final version of the RMF feature significantly lower maximum packet delays (10 s compared to 50 s) and significantly lower packet loss ratios (0% - 0.41% compared to 0.37% - 5.03%). Furthermore the average length of a congestion phase is reduced significantly by our ICSIC detector. While the average congestion phase for the pre RMF implementation ranges from 3300 ms to 4600 ms, the average congestion phase for all RMF measurements is less than 700 ms.

Secondly, an example trace shows that the ICSIC detector can not prevent ICSIC in all situations. The example identifies an arbitrarily introduced delay between two consecutive packets of a chirp of more than one second, which simply can not be detected prior to its arrival. Nevertheless the same example shows that the ICSIC detector acted as quickly as possible and ended the ICSIC phase as soon as possible.

Third, we give an example showing that the tail drop challenge is solved, too. And finally, a detailed overview on measurement results of one of the latest RMF traces completes the evaluation section.

We conclude that our presented measurement algorithm, which facilitates simultaneous high frequency measurements of key network characteristics of conventional mobile cellular networks while moving, indeed solves the identified measurement challenges. Furthermore, we released the RMF as source code on github[1] to allow anyone to gather their own cellular network traces. For the future we promise to release our complete tool chain allowing the simulation of *Car-to-X* (C2X) applications over mobile cellular networks.

4.2 Contribution

The first key contribution of this paper is the introduction of our enhanced measurement method allowing frequent measurements of application-to-application available data rates, loss rates and delays simultaneously for both communication directions. We especially focus on the measurement of cellular networks as the resulting traces are needed as input for our trace-based simulation model described in [1] (see Chapter 3). Therefore our measurement method is able to adapt to fast changing network conditions of mobile cellular networks and reduces the length of *inter-chirp self-induced congestion* (ICSIC).

The second key contribution of this paper is the evaluation of the implementation of this measurement method. Our evaluation proves that the identified challenges induced by the high measurement frequency, the combined delay and available data rate measurements and the rapidly changing network conditions are solved.

4.3 Personal Contribution

Elsevier announces a CiteScore of 2.97, an impact factor of 2.099, a 5-Year Impact Factor of 1.732, a Source Normalized Impact per Paper (SNIP) of 2.002 and a SCImago Journal Rank (SJR) of 0.889 for the Computer Communications Journal.

Norbert Goebel, the main author of this paper and the author of this thesis, recognized that current measurement methods do not suffice for trace-based cellular network simulation. Thus he started to design and develop the measurement method described in this paper. He supervised multiple master theses and student projects

[1] `https://github.com/hhucn/rmf`

working on the implementation of parts of the final outcome. The work on the implementation was started by Sebastian Wilken during his master thesis. Christian Lange, Tobias Krauthoff and Franz Kary extended the RMF by adding the safeguard channel, source-based routing and the kernel module helping to overcome the tail drops. Norbert Goebel chose to implement a bulk traffic measurement methodology and developed the techniques needed for the simultaneous measurements. Furthermore he developed and implemented the ICSIC detector presented in this paper, with Tobias Krauthoff giving valuable input in discussions. He also conducted all the measurements presented in the paper, conducted the evaluation and discussed them with Tobias Krauthoff, Kalman Graffi and Martin Mauve. Norbert Goebel wrote the paper with editorial help from Tobias Krauthoff, Kalman Graffi and Martin Mauve.

4.4 Importance and Impact on the Thesis

The paper summarized in this chapter solves the challenges cellular networks impose on simultaneous measurements of key network characteristics for both communication directions. It directly addresses and refines the challenges identified in our previous paper [1]. The paper introduces our measurement algorithm that is able to perform high frequency network measurements in fast changing network conditions like cellular networks. It further shows that no other currently available network measurement framework is designed to conduct these combined measurements and most of them cannot even cope with the fast changing network conditions. This paper shows that sound network traces of cellular networks can be gathered by anyone with affordable hardware. Therefore it underlines that trace-based cellular network simulations are feasible. Hence this paper is an essential building block of our simulation tool chain. The measurements conducted with the RMF introduced in this paper can be processed (as explained by our technical report [3] presented in Chapter 5) to form the simulation basis for our coupled V2X simulation framework published in [4] and presented in Chapter 6.

5 Aggregation and Map-Matching of Mobile Cellular Network Traces

This chapter summarizes the contributions and gives a verbatim copy of the technical report [3]:

> Norbert Goebel, Adrian Skuballa, Martin Mauve, Kalman Graffi:
> *Aggregation and Map-Matching of Mobile Cellular Network Traces*
> Published as a Technical Report at the Computer Science Department of the Heinrich Heine University Düsseldorf, July 2016.

Usually, traffic simulators operate on graph-based road networks, while position-based network traces log *Global Positioning System* (GPS) positions. Thus network traces have to be post-processed to transform the measured positions to the simulator's coordinate system. In addition, GPS positions are error prone and need to be map-matched onto the road network graph. Furthermore the network measurements normally do not timely match with the timestamps of the recorded GPS positions and need to be matched onto the road network graph, too.

The goal of our technical report [3] is to fill the post-processing gap between our mobile cellular network measurements conducted using the *Rate Measurement Framework* (RMF) [2] (see Chapter 4) and our trace-based simulation of *Vehicle-to-X* (V2X) applications explained in [4], presented in Chapter 6.

5.1 Paper Summary

This technical report introduces and explains our tool chain that post-processes our network traces gathered with the RMF (see Chapter 4) and generates output that can be used as a basis for our trace-based V2X simulations [4] presented in Chapter 6.

The key contributions of this technical report are:

1. The explanation of our methodology to map-match network traces generated by the RMF onto a road network graph defined by an OpenStreetMap XML file.

2. The introduction of multiple trace aggregation schemes that allow the combination of multiple network traces for trace-based cellular network simulations in a V2X context.

3. The presentation of a walk through of our complete post-processing work flow based on example measurements.

The RMF measures available data rates, delays and loss rates with up to 5 Hz. Simultaneously, other threads log the current position of the mobile measurement device using GPS at 1 Hz and status information of the cellular network modem with a frequency of 5 to 10 Hz. Each measurement is stored with its corresponding timestamp in nanoseconds. All this data has to be post-processed to be usable as a basis for our *Coupled Simulation of Mobile Cellular Networks, Road Traffic and V2X Applications using Traces* [4], which requires a graph based simulation database with tuples e,o,g,t,a,d,l,c. Each tuple stored in the database contains the **e**dge ID, the **o**ffset on the edge, the **g**roup ID, the measurement **t**ime, the **a**vailable data rate, the backbone **d**elay, the **l**oss probability and an ID representing the used **c**ell sector.

In the related work section we introduce the reader to the mapping data of *OpenStreetMap* (OSM) [8] and identify two different types of map-matching algorithms: *online* and *offline*. While *online* map-matching is conducted during the generation of the trace, it can only rely on current and past position measurements. *Offline* map-matching algorithms, which are usually used after a complete trace was collected, on the other hand, can utilize the whole trace to map-match any positions contained in the trace. As we do not rely on map-matching during trace generation — in contrast to for example navigation systems — we focus on offline map-matching algorithms. The paper outlines the basic concept of the chosen N-Route based *offline* map-matching algorithm presented in [50]. Finally, it summarizes our simulation framework introduced in [4] (see Chapter 6).

We chose to use the freely available mapping data of OpenStreetMap as basis for the road network graph, as the traffic simulator *Simulation of Urban Mobility* (SUMO) [51], used by our simulation environment, relies on it. Therefore, additionally to network traces generated by the RMF [2], our tool chain requires a genuine OSM XML file containing mapping data of the network measurement's region.

Our complete tool chain is divided into four Java programs with dedicated responsibilities: *OSMParser, JXMapMatchVer3, DataUnion* and *OSMJoin*. This allows easy reuse of output of each tool in different following programs in our tool chain.

In the first post-processing step, conducted by our *OSMParser*, a given OSM XML is condensed by filtering all data that is neither inside of the measurement area nor describes a road or its characteristics. Additionally, the OSMParser performs operations to assure that the later used *SUMO netconverter* does not merge non-identical nodes, which would result in errors in the road network-graph. The resulting output file is fed to *SUMO's netconvert* to generate the road-network graph which is used by SUMO in the *V2X Simulation Runtime Infrastructure* (VSimRTI) V2X simulations. This shrinking process reduces execution times for the second post-processing step, the map-matching, and for the traffic simulations using SUMO.

Our semi-automatic Java application *JXMapMatchVer3* uses a modified version of an algorithm presented in [50], which uses the *multiple hypothesis technique* (MHT) to estimate the route taken by the tracing vehicle. This algorithms usually produces close map-matching results. Nevertheless, flaws caused by bad GPS reception can lead to trace segments mapped to the wrong road segments. Thus our implementation features a manual correction mode, which allows to alter the map-matching result. In a final step, the map-matcher maps all measurements to the road network graph by interpolating their positions based on the individually logged timestamps.

The third post-processing tool described in the technical report is *DataUnion*. It is responsible for cell sector identification, the combination of multiple measurements and an optional selection of desired network characteristics without sacrificing simulation quality. It adds the corresponding *cell ID* to each position-based network characteristic in the simulation database during cell sector identification. Furthermore, it fills the *group ID* field in the simulation database while combining measurements of multiple network traces. The third option of DataUnion allows to keep only one measurement group (group ID) for each street section. We implemented a multitude of filters allowing to feature simulation needs for many different situations.

The fourth and last tool introduced by this paper is *OSMJoin*. It can be used to merge multiple map-matched routes into one OSM compatible XML file. This allows to combine multiple measurement traces for use in the simulation as basis for the road network graph used by SUMO.

Following the technical explanations of the complete tool chain, the technical report depicts an exemplary work flow of all the presented tools and by this shows how network traces generated by the RMF can be post-processed to build the simulation basis for our simulation environment.

The complete tool chain presented in this technical report is published under MIT License at github [1,2,3,4].

5.2 Contribution

The first key contribution of this technical report is the introduction of our map-matching methodology used to map our position-based cellular network traces onto an OSM defined road network graph.

Furthermore we explain how multiple network trace can be aggregated to form a combined, viable simulation base for our trace-based cellular network simulations in a V2X context.

Finally, the paper presents a walk through of our complete post-processing work flow on the example of real measurements.

5.3 Personal Contribution

Norbert Goebel, the author of this thesis and the main author of the paper, sighted different map-matching algorithms and evaluated which algorithm best fits the requirements to map-match our network traces onto an OSM road network graph. He supervised the implementation work on JXMapMatch, conducted by Tobias Amft, Daniel Elmo and Adrian Skuballa. The other tools — except SUMO's netconvert — presented in this paper were developed by Adrian Skuballa during his master thesis, which was supervised by Norbert Goebel. Norbert Goebel identified the need for the grouping of traces and suggested the use of a graph-based structure. Adrian Skuballa, Raphael Bialon and Norbert Goebel worked out the details of the method described in this technical report. Furthermore Norbert Goebel suggested that a selection process for multiple measurements per edge like for example "best average delay" would further enrich the post-processing. The implementation of this feature was conducted by Adrian Skuballa during his master thesis. This technical report was written by Norbert Goebel with editorial help by Adrian Skuballa, Martin Mauve and Kalman Graffi.

[1] https://github.com/hhucn/OSMParser
[2] https://github.com/hhucn/JXMapMatchV3
[3] https://github.com/hhucn/OSMJoin
[4] https://github.com/hhucn/DataUnion

5.4 Importance and Impact on the Thesis

The technical report summarized in this Chapter explains how position-based cellular network traces gathered with the RMF (see Chapter 4) can be map-matched and post-processed to a format usable by our trace-based simulation of cellular networks for V2X simulations (see Chapter 6).

This technical report thus closes the gap between the two major research topics of this thesis: The gathering of viable cellular network traces and the trace-based simulation of cellular networks. By describing the complete work flow and releasing our tool chain as open source at github, we assure that anyone can perform all necessary steps to transform RMF cellular network traces to the format used by our simulation framework.

Aggregation and Map-Matching of Mobile Cellular Network Traces

Norbert Goebel   Adrian Skuballa   Martin Mauve   Kalman Graffi

Heinrich Heine University, Düsseldorf, Germany
Computer Science Department
Computer Networks Group

TECHNICAL REPORT TR-2016-001
HEINRICH HEINE UNIVERSITY, DÜSSELDORF, GERMANY
COMPUTER SCIENCE DEPARTMENT

JULY 2016

Aggregation and Map-Matching of Mobile Cellular Network Traces

Norbert Goebel Adrian Skuballa Martin Mauve Kalman Graffi
Department of Computer Science, University of Düsseldorf, Germany
{goebel, mauve, graffi}@cs.uni-duesseldorf.de , adrian.skuballa@uni-duesseldorf.de

***Abstract*—The trace-based simulation of *Vehicle-to-X* (V2X) applications relies on sound traces in a format usable by the simulation framework. However traces normally use gps coordinates and traffic simulators operate on graph based road networks. Furthermore positions gathered using gps are error prone. Thus some post-processing of the traces has to be conducted to map-match and prepare them for the V2X simulation framework. In this paper, we introduce our post-processing techniques used to make traces generated by the *Rate Measurement Framework* (RMF) available to our trace-based simulation environment. Furthermore we present a walk-through usage example of our tool-chain.**

I. Introduction

A fundamentally important requirement for trace-based simulations are sound traces in a format usable by the simulation model. While high quality network traces of high volatile networks can be obtained by various means, they often need to be aggregated to be useful for the simulation model. Network traces for trace-based simulation of V2X communication add the challenge of map-matching, where *Global Positioning System* (GPS) coordinates need to be matched onto a road network graph.

We utilize the RMF [1] to obtain high quality network traces of mobile cellular networks while being on the road.

During each RMF measurement drive, the following data is collected and stored with corresponding timestamps in nanoseconds:

- Available data rate, drop rate and delay for each communication direction with a frequency of up to 5 Hz.
- The position of the measurement vehicle in 1 Hz intervals
- Modem status information regarding the current network condition, like the used network cell sector and the used frequency band, gathered with a frequency of 5 to 10 Hz.

The collected data has to be post-processed before it can be used as a basis for our simulation framework [2] for "Coupled Simulation of Mobile Cellular Networks, Road Traffic and V2X Applications using Traces". The simulation framework requires a graph based simulation database with tuples (e, o, g, t, a, d, l, c). Each tuple contains the **e**dge ID, the **o**ffset on the edge, the **g**roup ID, the measurement **t**ime, the **a**vailable data rate, the backbone **d**elay, the **l**oss probability and an ID representing the used **c**ell sector. Hence a major task is the matching of coordinates in latitude and longitude onto the road network graph, which uses edges and offsets to define positions. Thereby each edge has a start and an endpoint and each point in between can be defined using the offset. As a side condition, the edgeIDs have to be convertible on the fly during the simulation to those used by the traffic simulator *Simulation of Urban Mobility* (SUMO), which has its own naming convention and itself transform *OpenStreetMap* (OSM) XMLs to its format. As SUMO at that point adds many small edges for lane switching, which are not present in the real world, we can not use the SUMO net graph directly for the map-matching process.

At least the following two post-processing steps, which are described in the paper at hand, had to be implemented:

- map-match the error prone GPS positions onto a road network graph
- map the measured network characteristics to positions on the same road network graph

Further challenges solved in the paper at hand are:

- the combination of multiple measurement traces of the same area for the simulation
- finding representatives for measurement groups
- choosing specific network conditions for the simulation by selecting traces which fit user defined network characteristics as closely as possible

We start the remainder of this paper by presenting the related work in Section II, followed by a short summary of the prerequisites for the post-processing in Section III. Further on we explain the map-matching process and the aggregation of multiple traces in Section IV. We show how the map-matched gps-data is transformed into (road-edge, offset) tuples and how a minimal SUMO net-graph containing only edges with measurements is built for the simulation. In Section V we present an example of the usage of the complete post-processing tool-chain. We conclude the paper with Section VI.

II. Related Work

A. OpenStreetMap (OSM)

OpenStreetMap [3] is a free, community driven world map. It offers decent map coverage for most parts of the world. The map data is accessible as map tiles and as XML files, which define the underlying data used for map rendering. The main components of an OSM XML file are defined by *node* and *edge* tags. While a *node* tag can contain many attributes, we just use the unique *id* and the position given as *lat*itude and *lon*gitude. Each *way* tag can store multiple attributes, too, and the most important one is its unique *id*.

```
<node id="2409285517" lat="51.1622775"
   lon="6.8725373"/>
<node id="296343732" lat="51.1622536" lon
   ="6.8726286"/>

<way id="1011096">
  <nd ref="2409285517"/>
  <nd ref="296343732"/>
  <tag k="highway" v="secondary"/>
  <tag k="lanes" v="2"/>
  <tag k="lit" v="yes"/>
  <tag k="lit_by_gaslight" v="no"/>
  <tag k="maxspeed" v="50"/>
  <tag k="name" v="Benrather␣Schlossallee
     "/>
  <tag k="postal_code" v="40597"/>
</way>
```

Listing 1. OpenstreetMap tags example with stripped creationtime, version and usernames.

In addition, each way can have multiple *child* tags, which are either *node (nd)* tags or *key-value* tags. Every way representing a road segment has at least two *nd* tags, which reference *node ids* representing the consecutive course of the road's segments. It optionally stores multiple *key-value* tags defining characteristics like speed limit, number of lanes or street type. An example excerpt of an OSM XML file with only one way consisting of two nodes is shown in Listing 1. We stripped the creation timestamps, version information and user-names in the tags shown, as they are not relevant for the paper at hand.

B. Map-Matching

The process of determining the most probable path a vehicle has traveled on a road network while analyzing a series of GPS positions is called *map-matching*. While GPS positions, as described in [4], are error prone, many use cases like navigation systems rely on the knowledge of the whereabouts of the vehicle on the road network. Thus many different map-matching algorithms were developed in the last two decades. Navigation systems have to determine the current position of the vehicle using an *online* map-matching algorithm. Such algorithms can only use the current and previous GPS measurements for the calculation of the current position on the road network graph. This is the main reason why their estimation is especially error prone on intersections. *Offline* map-matching algorithms, on the other hand, are used after a complete GPS trace has been recorded. They thus have an omniscient view on the complete measurement series and can use all GPS positions for the map-matching of every measurement. This can significantly raise the quality of the estimated vehicle's path on the road network. As conducting the network measurements using the RMF does not rely on map-matched positions on the road network, we focused on *offline* map-matching algorithms. While many offline map-matching algorithms incorporate data from a dead reconning device, the authors in [5] presented an offline map-matching algorithm that only relies on GPS coordinates and a directed graph representing the road network. Their algorithm uses the *mutiple hypothesis technique (MHT)* and works by minimizing the sum of the euclidean distances of all GPS measurements to the estimated routes. Starting with the N closest links of the road network to the first GPS coordinate, it keeps evaluating the N best routes — those with the smallest cumulative errors — while matching additional GPS coordinates. At junctions, additional routes for every outbound edge is added to the list of routes. After a new calculation of the distances, the routes with the highest distances are removed from the list of best routes until only N routes are left. The authors have shown that their algorithm is fast and yields good matching results. Thus we opted for this algorithm in our implementation.

C. Cellular Network Simulation

Introduced in [6], *Trace Based UMTS Simulation* (TBUS) forms the base for our cellular network simulation model. We extended the simulation model with a fair cell share model and implemented it as an *Objective Modular Network Testbed in C++* (OMNeT++) module in [2]. Therein we also presented our complete simulation framework using the *V2X Simulation Runtime Infrastructure* (VSimRTI) as the simulation core, SUMO for traffic simulations, OMNeT++ with our library *libtbus* for the simulation of the cellular network and the *V2X application simulator* (VSimRTI_App). We used the simulation framework with measurements gained with the RMF introduced in [1] and the map-matching process described in the paper at hand to simulate an *Emergency Warning Application* (EWA) using *European Telecommunications Standards Institute* (ETSI) standards and an enhanced, mobile cellular network aware EWA version. Using the simulation results, we were able to show that ETSI-defined timeouts for EWA applications have to be altered for cellular network communications and that an offloading of traffic to the geoserver significantly lowers the bandwidth needed, without sacrificing traffic safety.

III. Prerequisites and Format of the Network Traces

As a prerequisite to our post-processing, at least one network trace has to be gathered using the RMF [1]. Each trace generates four files which are needed as input for the post-processing:

- *cellinfo.txt* contains a consecutive list of timestamped status information of the modem (like used network cell)
- *downstream-data.csv* contains timestamped network characteristics observed in the downstream direction
- *upstream-data.csv* contains timestamped network characteristics observed in the upstream direction
- *gpsd.log* contains logged gps data like position and time

Furthermore, a genuine OSM XML with the mapping data of the regions the measurements were conducted in is needed. In the remainder of this paper we call this file `genuine.osm.xml`.

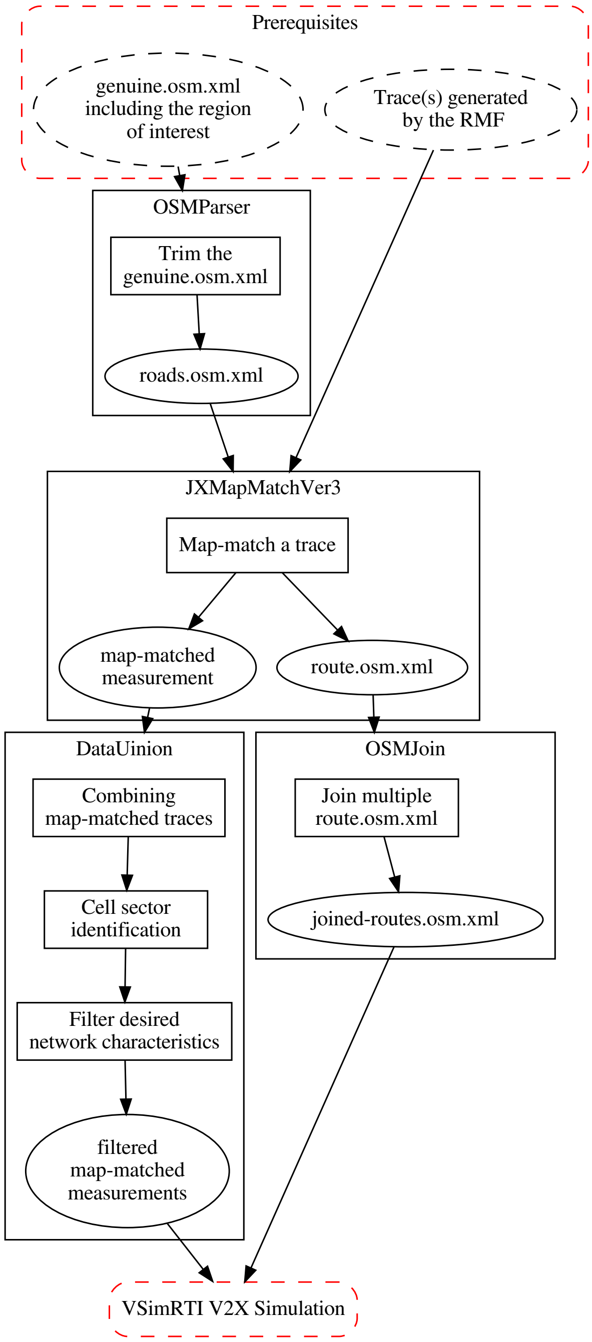

Figure 1. Overview of the post-processing work-flow.

The software needed for the implementation should run on any Linux or Windows platform. Working *java*, *sqlite3*, *SUMO* and (for the very last step) *VSimRTI* installations are required.

IV. Post-processing Steps

We split the post-processing of our network traces into multiple steps. This significantly reduces the workload if multiple measurements of the same area need to be processed and also allows flexible aggregation or selection of measurements. Figure 1 gives an overview about the multiple steps in our post-processing chain, which we describe in detail in the following subsections. The dashed red lines surrounding the prerequisites and the VSimRTI V2X simulation are not part of the post-processing work-flow. The post-processing itself is split into four major parts with headings naming the corresponding programs developed for the task at hand. Bubbles enclosed by rectangles mark steps in the work-flow. Round bubbles surround output files generated by the previous process.

A. *Trimming the OSM File (*OSMParser*)*

In the first post-processing step, the *OSMParser* is used to shrink a genuine OpenStreetMap XML file (`genuine.osm.xml`) by filtering all information, that is not related to roads and by removing all parts that do not belong to the measurement region or the region of interest. The region of interest hereby is a rectangular region defined by latitude and longitude ranges. These ranges can either be defined by hand through a command line switch, or they are extracted from the *gpsd.log* file(s) of measurements passed to the *OSMParser*. Furthermore, the *OSMParser* performs some operations to assure that a bijective function between OSM coordinates and the (edgeID, offset) tuple in the traffic simulator SUMO exists. As SUMO's *netconvert* merges nodes with the same coordinates, but different nodeIDs into one node, it might create intersections in the SUMO road-graph which are not present in the OSM data. Thus the *OSMParser* searches for nodes with identical latitude/longitude coordinates and increments the latitude of each but the first "duplicate" nodes in steps of 0.00000001 degrees (approximately one millimeter). Additionally, the *OSMParser* splits OSM ways consisting of more than two nodes into separate ways consisting of exactly two nodes to circumvent a length calculation problem of netconvert.

The output generated by the *OSMParser* is an OSM compatible XML-File, which we refer to as `roads.osm.xml`. It only consists of nodes and ways. The nodes have unique nodeIDs and distinct coordinates and are referred to by at least one way. The ways have unique wayIDs, too, and refer to exactly two of these nodes. Each edge is directed. An excerpt of a `roads.osm.xml` with a single way and its two nodes is shown in Listing 2.

The `roads.osm.xml` output of the *OSMParser* then is fed to SUMO's netconvert, which generates a corresponding SUMO road network graph `roads.net.xml` from it. An excerpt showing the SUMO edge generated from the way shown in Listing 2 is shown in Listing 3. It shows that netconvert uses the wayID and the nodeIDs defining the way to build the unique edgeID for SUMO's network graph.

This network graph is used for the traffic simulations by SUMO in the VSimRTI V2X simulations. As the nodeIDs and wayIDs can change in OSM over time (e.g. between releases) `roads.osm.xml` and `roads.net.xml` are only useful if used in conjunction. `roads.net.xml` files generated

```
<node id="1573051002" lat="51.1227132"
      lon="6.7750606"/>
<node id="1836570477" lat="51.1227595"
      lon="6.7750414"/>
<way id="1000596">
  <nd ref="1573051002"/>
  <nd ref="1836570477"/>
  <tag k="highway" v="residential"/>
  <tag k="maxspeed" v="30"/>
  <tag k="name" v="Am␣Schwimmbad"/>
  <tag k="old_way_id" v="23645367"/>
  <tag k="oneway" v="yes"/>
  <tag k="old_oneway" v="yes"/>
</way>
```

Listing 2. Excerpt of a `roads.osm.xml`.

```
<edge id="1000596
    _1573051002_1836570477_1573051002"
   from="1573051002" to="1836570477" priority=
       "-1">
  <lane id="1000596
      _1573051002_1836570477_1573051002_0"
      index="0" speed="8.33" length="4.02"
      shape="2843.76,42.01␣2842.87,45.93"/>
  <lane id="1000596
      _1573051002_1836570477_1573051002_1"
      index="1" speed="8.33" length="4.02"
      shape="2840.55,41.27␣2839.65,45.20"/>
</edge>
```

Listing 3. Excerpt of a `roads.net.xml`.

using a different `roads.osm.xml` are not compatible for the ongoing map-matching and simulation process. But if the measurement and simulation area does not change, the generated `roads.osm.xml` and `roads.net.xml` files can be used for multiple map-matching processes and simulations even if different traces are used.

The significant size reduction of the `genuine.osm.xml` file reduces loading and execution time of the actual map-matching process and the simulation.

*B. Map-Matching (*JXMapMatchVer3*)*

Given that a network trace and corresponding `roads.osm.xml` and `roads.net.xml` files exist, the second post-processing step, the actual map-matching, can be carried out. We developed a semi automatic graphical Java Application called *JXMapMatch* to do the actual map-matching. In a first step, it map-matches every collected GPS position (*gpsd.log*) onto the road network graph described by the `roads.osm.xml` using a modified version of the algorithm introduced in [5] and explained in Section II. After the algorithm has calculated the most probable route taken by the measurement vehicle, the graphical editor allows manual changes of the route. This enables an easy correction of map-matching errors caused by bad gps accuracy or faulty mapping data. Finally, the actual measurement data is matched onto the road network graph by interpolating the position based on the timestamps of each measurement

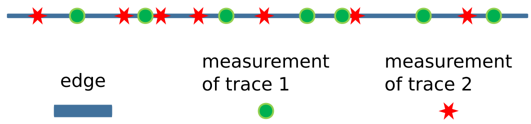

Figure 2. Example of an edge that is passed two times.

and the map-matched route taken by the vehicle. Each map-matching process using *JXMapMatch* generates multiple output files:

1) multiple KML files are generated, which allow the easy visualization of the measurements using Google earth [7]
2) map-matched measurements (`map-matched.csv`)
3) an even smaller version of the `roads.osm.xml` file containing only the roads used in the map-matched traces is generated, which we refer to as `route.osm.xml`

*C. Identifying Network Cell Sectors and Combining Multiple Measurements (*DataUnion*)*

The third post processing step, implemented as *DataUnion*, addresses two challenges:
1) it identifies the different cell sectors of the mobile cellular network
2) and it allows the combination of multiple measurements

1) Identifying Network Cell Sectors: For the trace-based simulation it is important to know the cellular network cell sector the tracing vehicle was communicating with at a specific time. As this information is not readily available, it must be derived from the logged modem information. The RMF logs the *Location Area Code (LAC)* and the *CellID* when communicating with a 2G cellular network cell and, when using a 3G network connection, the *Channel* and the *Primary Scrambling Code (PSC)*. On the one hand, two neighboring network cells must not use the same settings, as their communication would interfere. On the other hand, the number of combinations of LACs and CellIDs respectively channels and PSCs is limited. Thus multiple cells with the same settings exist. *DataUnion* therefore first groups measurements with the same connection settings. Afterwards it checks the distances of the measurements in any group and forms subgroups if measurements are farer apart than a parameterized distance. We set the default for this distance to 5 km as mobile cellular network cells can be quite large and the 5 km distance worked well in all our traces. Each such subgroup is assigned a unique CellID by *DataUnion*, which is associated with the corresponding mobile cellular network characteristics.

2) Combining Multiple Measurements: Often the map-matched measurements of multiple traces need to be combined to enlarge the simulation area, whereas in other situations it is desirable to aggregate multiple measurements of the same area.

Figure 2 sketches an example situation where an edge is passed two times during measurements. Merging the traces of Figure 2 based on their offset on the edge would lead to the sequence 21212212112121. Imagine two different situations for the measurements where the first pass was conducted during rush hour and the second pass was made in the middle of the night. Obviously, the measured network characteristics of these traces can deviate largely. As a consequence, multiple switches between extremely different network characteristics are possible. This can have a significant influence on the simulation outcome and lead to unrealistic network simulations. Hence *DataUnion* instead allocates a unique *groupID* g to every measurement. Thereby the groupID g is equal for all measurements mapped to the same edge and belonging to the same passing of the edge. For every network measurement the post-processing thus results in the desired tuple (e, o, g, t, a, d, l, c) required by our simulation model.

*D. Selecting desired network characteristics (*DataUnion*)*

A commonly asked question when analyzing algorithms target the best, average and worst case scenarios. Defining equivalent realistic cases for cellular network communication simulations is quite challenging as the same thoughts disallowing easy combination of multiple measurements have to be applied, too. Our solution for this is a two step filter option build into *DataUnion*. It allows the selection of a specific groupID per edge for the simulation. In the first step, for each group of measurements sharing the same groupID, a representative value either for the delay, the available data rate or the loss rate is selected as:

- **min**: the minimal value of the group
- **max**: the maximal value of the group
- **avg**: the average of the group
- **med**: the median of the group
- **minstde**: the maximum of "the average value subtracted by the standard deviation" and the "min" value
- **maxstde**: the minimum of "the sum of the average value and the standard deviation" and the "max" value
- **avgstde**: the arithmetic average of all values within the range of minstde and maxstde. We specifically added this option to minimize the impact of measurement outliers as often only 3 to 10 measurements are forming a value group.
- **ww**: the way weighted average of the value group. In the simulation, a measurement is used from its measured position until a position with another measurement is passed. Thus this representative selection weights the measurements depending on the length their value is valid on their edge.

In the second step, a single groupID per edge is selected based on the representative and one of the following, selectable algorithms:

- **all**: no filtering is performed
- **med-**: the groupID with the median representative value is chosen for each edge. If two medians exist, the first is chosen.
- **med+**: the groupID with the median representative value is chosen for each edge. If two medians exist, the second is chosen.
- $p \in [0, 1]$: the groupID with the representative closest to $minrep + p \cdot (maxrep - minrep)$ is chosen.

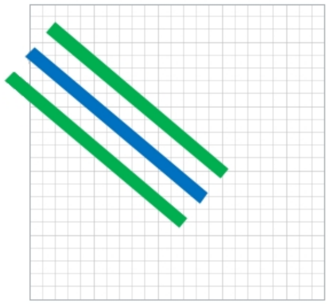

Figure 3. SUMO creating negative coordinates for one lane (green) from the blue edge.

After these two filter steps exactly one groupID is chosen per edge as long as not "all" is selected. All generated tuples (e, o, g, t, a, d, l, c) are stored in the graph-based simulation database. As switches between traces using this algorithm only occur when changing edges, it is possible to choose the network characteristics that are most interesting for the desired simulation scenario, but still retain realistic network simulations.

*E. Joining multiple `route.osm.xml` files (*OSMJoin*)*

While the measurements itself are combined and aggregated by *DataUnion*, the `route.osm.xml` files corresponding to these traces need to be merged, too, as the traffic simulator SUMO needs to know the ways and nodes of all routes taken by any of the combined traces. We thus implemented *OSMJoin*, which allows the joining of multiple `route.osm.xml` files while keeping the same nodeIDs and wayIDs used in the `roads.net.xml` used by SUMO. Furthermore, *OSMJoin* helps to avoid a simulation bug caused by SUMO, which transforms the geographical coordinates of the given `roads.net.xml` to its own coordinate system consisting of positive numbers only. During the creation of the internal route graph, SUMO converts OSM ways with multiple lanes into multiple parallel ways with one lane. This, as shown in Figure 3, can lead to negative coordinates, resulting in severe problems during the VSimRTI simulation. Figure 4 shows how *OSMJoin* solves this problem by adding two single lane horizontal ways above and below the area defined by the given `route.osm.xml` file(s). These two ways are not connected to any other way and the coordinates of the defining nodes are lower/higher than the coordinates of all nodes. The used offset is parameterized. This solves the negative coordinates problem in the simulation and also assures that the two new ways are not used during the simulation, as each route in SUMO needs to be at least two ways long.

Afterwards, the `joined-routes.osm.xml` is used by netconvert to create the SUMO road network graph `joined-routes.osm.xml`.

Figure 4. OsmJoin generated minimal OSM graph with top and bottom single lane ways (Map ©OpenStreetMap [3])

V. Exemplary Workflow

In this section we show an exemplary workflow of our post-processing tool-chain, which is available for download at github [1,2,3,4]. All listings used in this section show single line commands executed in a Linux bash shell. To enhance readability we manually split the commands into multiple lines — one line for each parameter. We also assume that the called program either resides in the current directory or in the path. Furthermore, the input files are accessible in subdirectories to the current working directory in our examples following a naming scheme of *YYYYMMDD/HHMMSS*. In the shown examples we combine four traces conducted on 3rd August 2015.

A. OSMParser and netconvert

The OSMParser is used to shrink a `genuine.osm.xml` file by deleting all non-road information and by constraining to the area of interest. In the following example the OSMParser parses a genuine OSM XML file (Parameter *-oi*) of the City of Düsseldorf and its surroundings. The name of the output file is defined by *-oo* and all gpsd.log files preceded by the *-g* parameter are used to constrain the area of interest to the rectangle formed by (minlon,minlat) and (maxlon,maxlat) contained in the gps logs.

```
java OSMParser
    -oi duesseldorf-regbez-latest.osm.xml
    -oo roads.osm.xml
    -g 20150803/090007/gpsd.log
    -g 20150803/091227/gpsd.log
    -g 20150803/092532/gpsd.log
    -g 20150803/093139/gpsd.log
```

[1]https://github.com/hhucn/OSMParser
[2]https://github.com/hhucn/JXMapMatchV3
[3]https://github.com/hhucn/OSMJoin
[4]https://github.com/hhucn/DataUnion

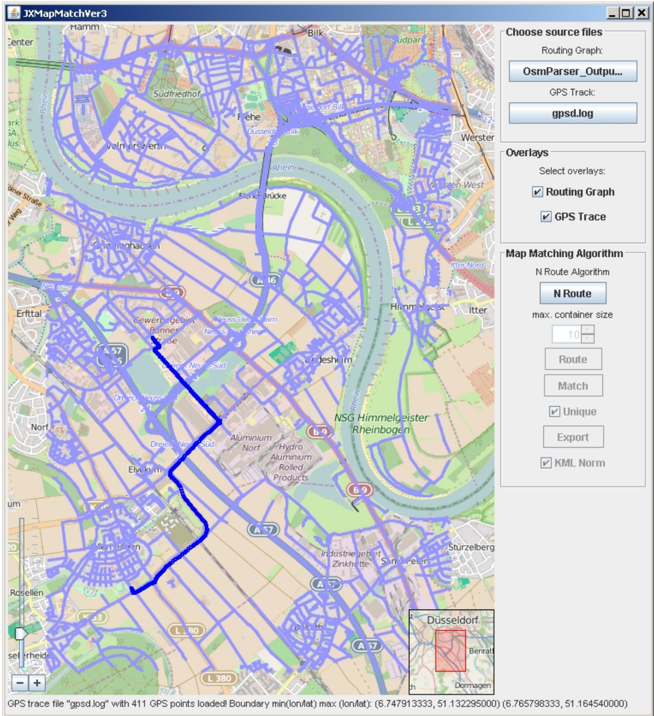

Figure 5. Screenshot of JXMapMatchVer3 (Map ©OpenStreetMap [3]).

Even though the multi-node ways of the genuine OSM XML file (303MB in this example) are split to multiple two-node ways in the output file, the output file is significantly smaller (<3MB).

As stated in Section IV-A, the generated `roads.osm.xml` needs to be converted to a `roads.net.xml`. This can be accomplished by calling netconvert:

```
netconvert
  --osm-files roads.osm.xml
  -o roads.net.xml
```

B. Map-Matching with JXMapMatchVer3

In Section IV-B, we explained how the map-matching process works. As *JXMapMatchVer3* is a GUI-based application, no command line parameters are needed. Thus we give a short description of the workflow using the screenshot shown in Figure 5. The GUI is split into a MapView on the left and a navigation pane on the right. The MapView shows OpenStreetMap Map Tiles in the background and (when `roads.osm.xml`, `roads.net.xml` and a gpsd.log are loaded) the road network graph (lilac) and the logged gps positions (blue).

C. Aggregating map-matched measurements with DataUnion

After several traces have been map-matched by *JXMapMatchVer3* using identical `roads.osm.xml` and `roads.net.xml` files, their traces can be combined by *DataUnion*. As explained in Section IV-D, the data can be combined leaving multiple groupIDs per edge. An exemplary command is:

```
java DataUnion
    -o combined-map-matched.csv
    -g all
    -i 20150803/090007/map-matched.csv
    -i 20150803/091227/map-matched.csv
    -i 20150803/092532/map-matched.csv
    -i 20150803/093139/map-matched.csv
```

The aggregation of the same traces with active filtering could look like:

```
java DataUnion
    -o filtered-map-matched.csv
    -g med-
    -t datarate
    -r ww
    -i 20150803/090007/map-matched.csv
    -i 20150803/091227/map-matched.csv
    -i 20150803/092532/map-matched.csv
    -i 20150803/093139/map-matched.csv
```

In this example, for each edgeID with multiple assigned groupIDs only one groupID is saved in the `filtered-map-matched.csv` output. The selection bases on the data rates. The representative for each group is the way weighted average of the group. The group with the lower median is selected as the only group for this edge in the filtered output.

D. Joining multiple `route.osm.xml` by OsmJoin

A usage example for *OsmJoin*, which combines the routes of four traces to `joined-routes.osm.xml` is:

```
java OsmJoin
    -o joined-routes.osm.xml
    -i 20150803/090007/route.osm.xml
    -i 20150803/091227/route.osm.xml
    -i 20150803/092532/route.osm.xml
    -i 20150803/093139/route.osm.xml
```

E. Generate Necessary Files for the Simulator

Finally, the aggregated measurements need to be converted to a sqlite database which is used by the OMNeT++ module of the simulation framework. Let the measurements be contained in `filtered-map-matched.csv`, then the *sqlite* database can be created with the commands:

```
awk --field-separator="," 'BEGIN{i=0} /down/ {
    print i "," $24 "," $3 "," $11 "," $12 ","
     $23 "," $15 "," $17 "," $16; i=i+1}'
    filtered-map-matched.csv > filtered-
    download.csv
awk --field-separator="," 'BEGIN{i=0} /up/ {
    print i "," $24 "," $3 "," $11 "," $12 ","
     $23 "," $15 "," $17 "," $16; i=i+1}'
    filtered-map-matched.csv > filtered-upload
    .csv
cat << EOF | sqlite3 test_edge.sqlite
pragma user_version=1;
.separator ,
.import filtered-download.csv download
.import filtered-upload.csv upload
EOF
```

Furthermore, the `joined-routes.osm.xml` needs to be prepared for SUMO and VSimRTI, which we use for our V2X simulations in [2]. For the generation of the SUMO net file we use netconvert with the following parameters:

```
netconvert
  --osm-files joined-routes.osm.xml
  -o joined-routes.net.xml
```

The conversion for VSimRTI is performed by *scenario-convert*, which is distributed with VSimRTI. We used VSimRTI Version 0.15 for our simulations, and thus used the following command to create the database:

```
java -jar scenario-convert-0.15.0.jar
    --osm2sumo
    -d joined-routes.db
    -i joined-routes.osm.xml
    -n
```

VI. CONCLUSION

In this paper we introduced the methods used to post-process our position-based mobile cellular network traces generated using the RMF. The post-processed output builds the simulation basis for our V2X simulation framework presented in [2]. We explained how the conversion from gps positions to positions in a road network graph works and introduced our map-matching implementation. Furthermore, we introduced our aggregation approach for multiple traces by a multitude of filtering options. In Section V we presented a walk through example of the complete tool chain. The complete post-processing tool chain presented in the paper at hand is available as sources (see Section IV) under MIT licenses. The results of the post-processing are already actively used in V2X simulations with the use of VSimRTI and our OMNeT++ module presented in [2].

REFERENCES

[1] N. Goebel, T. Krauthoff, K. Graffi, and M. Mauve, "Moving Measurements: Measuring Network Characteristics of Mobile Cellular Networks on the Move," in *2015 IEEE Vehicular Networking Conference (VNC) (IEEE VNC 2015)*, Kyoto, Japan, Dec. 2015.

[2] N. Goebel, R. Bialon, M. Mauve, and K. Graffi, "Coupled simulation of mobile cellular networks, road traffic and V2X applications using traces," in *IEEE ICC 2016 - Mobile and Wireless Networking Symposium (ICC'16 MWN)*, Kuala Lumpur, Malaysia, May 2016.

[3] OpenStreetMap, "OpenStreetMap," [Accessed Feb. 14th 2016], Copyright and License: www.openstreetmap.org/copyright. [Online]. Available: http://www.openstreetmap.org/

[4] S. Brakatsoulas, D. Pfoser, R. Salas, and C. Wenk, "On map-matching vehicle tracking data," in *Proceedings of the 31st International Conference on Very Large Data Bases*, ser. VLDB '05. VLDB Endowment, 2005, pp. 853–864. [Online]. Available: http://dl.acm.org/citation.cfm?id=1083592.1083691

[5] F. Marchal, J. Hackney, and K. W. Axhausen, "Efficient Map-Matching of Large GPS Data Sets - Tests on a Speed Monitoring Experiment in Zurich, volume 244 of Arbeitsbericht Verkehrs und Raumplanung," Tech. Rep., 2004. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/citations?doi=10.1.1.113.4605

[6] N. Goebel, M. Koegel, M. Mauve, and K. Graffi, "Trace-based simulation of c2x-communication using cellular networks," in *11th Annual Conference on Wireless On-demand Network Systems and Services (WONS 2014) - Special Session on VANETs and ITS3*, Apr. 2014, pp. 108–115. [Online]. Available: http://dx.doi.org/10.1109/WONS.2014.6814730

[7] Google, "Google earth," August 2007. [Online]. Available: http://earth.google.de/

6 Coupled Simulation of Mobile Cellular Networks, Road Traffic and V2X applications using Traces

This chapter summarizes the contributions of the paper [4]:

> Norbert Goebel, Raphael Bialon, Martin Mauve, Kalman Graffi:
> *Coupled Simulation of Mobile Cellular Networks, Road Traffic and V2X applications using Traces*
> In Proceedings of the IEEE International Conference on Communications - Mobile and Wireless Networking Symposium (ICC'16 MWN), Kuala Lumpur, Malaysia, May 2016.

The development of *Vehicle-to-X* (V2X) applications is time intensive and costly as real-world tests require the coordination of multiple vehicles, persons and infrastructure. This is especially true when a cellular network and not the competing IEEE 802.11p *Wireless Local Area Network* (Wi-Fi) is used as the communication technology.

A common approach to lower development costs and speed up the development process for V2X applications typically involves simulation. While multiple well-established network simulators support the simulation of IEEE 802.11p Wi-Fi, simulation models for cellular networks that do not rely on confidential *Non-Disclosure Agreement* (NDA) knowledge of the to be simulated cellular network are mostly probabilistic and thus their results might deviate largely from the real world.

The goal of our paper [4] thus is to close this simulation gap and show that trace-based simulations of cellular networks for V2X simulations are feasible.

6.1 Paper Summary

This paper introduces our complete simulation environment allowing coupled simulation of mobile cellular networks, road traffic and V2X applications using cellular network traces.

Its key contributions are:

1. The introduction of our complete simulation environment for V2X applications using cellular network communication.

2. The introduction of our trace-based simulation model as an *Objective Modular Network Testbed in C++* (OMNeT++) module usable by any simulation relying on OMNeT++ [52] for network simulations.

3. The demonstration of the usability of our simulation environment on the example of the *European Telecommunications Standards Institute* (ETSI) defined *Emergency Warning Application* (EWA) which additionally underlines the significance of simulations when porting IEEE 802.11p V2X applications for cellular network use.

As foundation, the paper depicts the used simulation environment *V2X Simulation Runtime Infrastructure* (VSimRTI) and explains how the different simulators are coupled by VSimRTI. Additionally, the paper gives a brief description of the simulators used in our simulations: The traffic simulator *Simulation of Urban Mobility* (SUMO), the network simulator OMNeT++ and the *V2X application simulator* (VSimRTI_App).

Furthermore, we explain how we gather cellular network traces and how we map-match the traces onto a road network graph. We thereby give short summaries of the measurement paper [2] and the map-matching technical report [3] presented in Chapters 4 and 5 of this thesis. These map-matched cellular network traces form the basis of our simulation model introduced in [1] (see Chapter 3).

The paper summarizes our simulation model [1] and introduces the OMNeT++ cellular network simulation module built upon this model. Furthermore the *fair share* cell share model is introduced in this section. It enhances our simulation model and is responsible for distributing the available data rate among competing devices in the same network cell sector.

Based on these works we explain why we chose the ETSI defined Emergency Warning Application as our example use case to demonstrate the usability of our simulation environment. This V2X application periodically broadcasts warning messages to inform vehicles in the surrounding of an emergency vehicle of its presence. It thus raises the awareness of the drivers of vehicles in the vicinity of an emergency vehicle. In anticipation of the evaluation results we furthermore describe an enhanced, cellular network aware, EWA application that offloads some of the work and network load from the emergency vehicle to the *geoserver*, which normally monitors vehicle positions and coordinates geo-broadcast for V2X applications using cellular networks.

For the evaluation we collected and post-processed network traces of an industrial area. This data is used as simulation basis for the simulation of the EWA and enhanced EWA V2X applications in the simulation environment. The main focus of the evaluation is the combined message delay of each warning message.

In contrast to IEEE 802.11p communication, each warning message has to traverse the cellular network two times. First it has to pass the upstream from the emergency vehicle to the *geoserver* and then it has to pass the downstream connection from the *geoserver* to each receiving mobile node. Our simulation results indicate that the ETSI defined EWA has major difficulties in areas with low available data rates in the upstream and shows message delays of up to 70 s. The simulations also show that the improvements implemented in the enhanced EWA significantly lower the combined message delays, which stay below 5 s during the complete simulation. Additionally, the evaluation addresses the question, if the ETSI defined upper bound of 100 ms for message delays is feasible for cellular network usage and concludes that it should be raised. In our simulations an upper bound of 600 ms seems promising as more than 99% of the combined message delays for the enhanced EWA were below this bound. In contrast only 48% of the combined message delays of the unmodified, ETSI defined, EWA were below the 600 ms bound.

The paper concludes that further enhancements on the cell share model are possible and suggests to evaluate the impact of the utilized measurement traffic on the simulation.

Making extensive use of the work introduced in the other papers of this thesis (Chapters 3 to 5) [1], [2], [3], the paper at hand clearly shows the benefits of trace-based cellular network simulation for V2X application development and demonstrates its importance for V2X application simulation.

6.2 Contribution

The primary key contribution of this paper is the introduction of our complete simulation environment for V2X applications. Using cellular network traces as its simulation basis, it is capable of simulating V2X applications and their cellular network communications. Furthermore, it allows the coupled simulation of cellular network communications, road traffic and V2X applications using well-known simulators.

The second key contribution is the introduction of our trace-based simulation model as an OMNeT++ implementation usable by any simulation relying on OMNeT++ [52] for network simulations. Its sourcecode and installation instructions to directly include it in VSimRTI Version 0.15.0 are available on the authors' website[1] respectively links on that site.

Another key contribution is the evaluation of our implementation of the ETSI defined EWA. We use this V2X application to demonstrate the usability of our simulation tool chain. Furthermore, we outline its value when porting existing IEEE 802.11p V2X applications to cellular network communication.

6.3 Personal Contribution

The IEEE International Conference on Communications (ICC) 2016 received 2,469 paper submissions for the conference. 960 of these papers have been accepted for presentation and publication in the Conference Proceedings, resulting in an acceptance ratio of less than 39%.

Norbert Goebel developed and designed the simulation model that Raphael Bialon implemented as an OMNet++ module. Raphael Bialon also conducted the implementations needed to run the simulations using Fraunhofer Fokus VSimRTI. Norbert Goebel gathered and post-processed the cellular network traces using the *Rate Measurement Framework* (RMF) introduced in [2] that form the simulation basis of this paper. The measurements were Map-Matched and aggregated by Adrian Skuballa as described in [3]. The simulations were implemented, conducted and evaluated by Raphael Bialon and Norbert Goebel. Both also worked on the packaging and installation routines available for download on the institute's website that allow easy reproducibility of the simulations presented in this paper. The main part of the paper was written by Norbert Goebel. Raphael Bialon supplied the sections on the "cell share model", "VSimRTI and OMNeT++ modifications" and the related work section on "Coupling of Traffic and Network Simulation". Martin Mauve and Kalman Graffi assisted the writing process with editorial work.

[1] `https://wwwcn.cs.uni-duesseldorf.de/software/simulation.html`

6.4 Importance and Impact on the Thesis

The paper summarized in this chapter is the culmination of this thesis. It builds upon the work of all papers presented in Chapters 3 to 5.

It demonstrates the feasibility of trace-based simulation of cellular networks for V2X communication simulation by using our complete work flow from cellular network measurement via post-processing and map-matching of the network traces to coupled simulation of mobile cellular networks, road traffic and V2X applications.

Using an exemplary V2X application, the ETSI defined Emergency Warning Application, the paper outlines the importance of a V2X simulation environment capable of realistic cellular network communication simulation for the development and evaluation of V2X applications.

We provide all the sourcecode of all software developed by us and used in the papers presented in this thesis (measurements, post-processing, map-matching and the OMNeT++ simulation model) and thus enable anyone to use our simulation tool chain and to build upon our work.

7 Conclusion and Future Work

Today's growing road traffic densities challenge the automotive industry to invent new safety features for their vehicles to reduce the number of accidents or at least to reduce their severity. Previous inventions rely on information generated by sensor in the vehicle itself and thus are limited to the line-of-sight. The advancements in wireless communication in the last two decades make message exchange between vehicles, which is known as *inter-vehicular communication* (IVC), *Vehicle-to-X* (V2X) communication or *Car-to-X* (C2X) communication, possible. In combination with position information obtained from systems like *Global Positioning System* (GPS), a new category of safety features is possible as V2X applications can combine information of other vehicles with their own sensor data to enrich their knowledge of the surrounding world.

The development of V2X applications and more so the evaluation of the impact of a specific V2X application is time intensive and expensive, as *Field Operational Tests* (FOTs) require a large amount of resources and time. Simulation offers a possibility to significantly reduce these costs and lower development times. While multiple network simulators allow realistic simulation of *Institute of Electrical and Electronics Engineers* (IEEE) 802.11p *Wireless Local Area Network* (Wi-Fi), the simulation of the second available V2X communication technology, cellular network communication, is much harder. Thus existing simulators either require confidential information on the cellular network to allow realistic simulations, or the simulation model is highly probabilistic. We aim to close this cellular network simulation gap with this thesis.

The goal of this thesis thus is to evaluate the feasibility of trace-based simulation of cellular networks for V2X simulations and to develop, evaluate and release a trace-based cellular network simulation model and couple it with a traffic simulator and a V2X application simulator.

Pursuing these goals, a multitude of questions in the research areas network measurements, map-matching, cellular network simulation and the coupling of multiple simulators arise. In the following, we conclude our findings in these research fields and summarize our contributions to them.

7.1 Conclusion

Chapter 1 introduces and motivates the concept of V2X communication to enhance road traffic safety and efficiency. We outline why cellular network simulations are complicated, but urgently needed to reduce the costs and speed up the development process for V2X applications. We introduce and propel the idea of a trace-based cellular network simulator to satisfy the simulation needs. Furthermore, we propose open research questions in four fields of research that have an impact on the goals of this thesis. The chapter briefly summarizes our contributions answering most open research questions and presents the outline of this thesis.

In Chapter 2 we introduce the related work of two major research fields which are not intensively covered in the papers presented in this thesis. First, we present multiple cellular network simulation approaches and explain their differences to our proposed trace-based simulation model introduced in Chapter 3 and enhanced in Chapter 6. Then we summarize the related work on the coupling of at least a network simulator and a road traffic simulator and evaluate the possible use of this related work for our research goals. We conclude this Chapter by explaining which approaches presented in the related work are used in our researches presented in this thesis.

In the introduction (see Chapter 1) we identified numerous research questions in four research fields that are of special interest to us as the majority of them needs to be answered to reach the goals of this thesis. In the process of answering these questions, two papers, one journal article and one technical report originated, which we present in this thesis. In the following, we outline these contributions.

- In Chapter 3 we examine the general feasibility of trace-based cellular network simulation for V2X simulations. The paper presented in this chapter introduces and evaluates our trace-based simulation model and our basic bulk-traffic measurement methodology. Comparing simulation results and real-world measurements, it demonstrates the feasibility of trace-based cellular network simulation. The paper presented in this chapter directly tackles many research questions in the network measurement and cellular network simulation sectors. In its evaluation section open challenges for our network measurement methodology are identified and postponed for future work. The presented paper is fundamental for this thesis as it proves the feasibility of trace-based cellular network simulations and identifies future research questions that we pick up in our other contributions.

- Consequently, the journal paper presented in Chapter 4 tackles the research challenges for the cellular network measurements identified in the paper presented in Chapter 3. The paper introduces our *Rate Measurement Framework* (RMF) that incorporates multiple feedback mechanisms to deal with the challenging fast changing network conditions cellular networks feature. The presented RMF allows the simultaneous, high frequency measurement of available data rates, loss rates and one-way delay for both communication directions between a mobile node and an Internet homed server. The network traces gathered with the RMF build the basis for our simulation model and are the input for our map-matching tool chain presented in Chapter 5.

- The technical report presented in Chapter 5 demonstrates the map-matching and aggregation of network traces gathered with the RMF onto a road network graph originating from *OpenStreetMap* (OSM) mapping data. Furthermore, it explains how the map-matched network traces are transformed into the graph-based structure required by our cellular network simulation model introduced in the paper presented in Chapter 6. The technical report introduced in this chapter thus closes the post-processing gap between our cellular network measurements and our cellular network simulations.

- The paper presented in Chapter 6 is the culmination of this thesis. It directly builds upon the work presented in Chapters 3 to 5. The paper again underlines the feasibility and usability of trace-based simulation of cellular networks for V2X simulations. First, it introduces an enhanced version of our trace-based cellular network simulation model as an *Objective Modular Network Testbed in C++* (OMNeT++) module and thus allows anyone to incorporate our simulation model into existing or new network simulations with the OMNeT++ framework. Additionally, it introduces and evaluates our complete simulation environment allowing coupled road traffic, V2X application simulation and cellular network simulation by using RMF traces in our OMNeT++ cellular network simulation model. And finally, the paper presents simulation results of an example V2X application initially defined for IEEE 802.11p Wi-Fi and our enhanced, cellular network optimized version of the same V2X application. On the one hand, these simulations demonstrate the usability of our complete simulation tool chain starting from the measurements up to the coupled simulation. On the other hand, the evaluation of these simulations underlines the importance of simulations, especially when porting applications defined for IEEE 802.11p to cellular network usage.

In conclusion, this thesis aims at improving the cellular network simulation situation in V2X simulations by introducing trace-based cellular network simulations. Pursuing this goal, numerous research questions in the fields of network measurements, map-matching, cellular network simulation and coupled simulation of road traffic, cellular networks and V2X applications are stated in this thesis and answered by our work presented in Chapters 2 to 6.

By publishing the sourcecode of all software developed by us that is introduced in the papers, the journal article and the technical report presented in this thesis, everyone can conduct V2X simulations using cellular network traces and anyone can build upon our work.

7.2 Future Work

During our research we identified numerous open research questions. While we answer plenty of them with this thesis, some still remain unanswered and we present them in this section as possible future work.

7.2.1 Verifying and Refining the Cell Share Model

In Section IV.A of our paper [4] presented in Chapter 6 of this thesis, we introduce an intermediate processing layer, the *cell share model*, which is responsible for the distribution of the measured available data rate to all nodes participating in the same network cell during the simulation.

The simulations presented in this paper use our *fair share* model, which, given that $n > 1$ vehicles are actively transferring data in the same cell, simulates the available data rate for each of these i vehicles as $A_{simulation}(i) = \frac{A_{measured}(i) \cdot 110\%}{n}$. Where $A_{measured}(i)$ is the available data rate measured for the current position of vehicle i.

We opted for the factor of $\frac{110\%}{n}$ instead of $\frac{100\%}{n}$ as our measurements typically are conducted with only one measurement device present in same network cell. Thus this one measurement device is competing with an unknown number of other devices for the available capacity. A measurement with multiple of our measurement devices per cell would most probably lead to lower available data rate measurements for each of our devices alone, but the added up available data rates of all our measurement devices would probably exceed those measured with a single measurement device. This is caused by the higher number of our devices competing with the same number of foreign equipment for the available data rate.

Certainly, the factor of $\frac{110\%}{n}$ is somewhat arbitrary and should be scientifically verified or refined. For this purpose we planned and partially conducted measurements with multiple measurement devices per cell with varying distances and *Signal-to-Noise Ratios* (SNRs). Our intention was to statistically analyze the measurement results and derive a sophisticated cell share model that we planned to implement in our simulation model. While the measurement results looked promising, time constraints impeded a scientifically sound result. Thus we emphasize that further refinement of the cell share model is desirable.

7.2.2 Enhancing Network Traces with Support of Modem Manufacturers

A common challenge when working with cellular network modems is the ability to gain useful status information from them. The cellular modems surely know the details of their connection status like the used spreading code (in UMTS), the used *Forward Error Correction* (FEC) code, the SNR, the *Block Error Rate* (BLER) or if it is currently being switched to another technology or network cell, as this information is used to participate in the cellular network. Still it is hard to obtain this information from the cellular modems. Often the manufacturers do not offer a dedicated interface for information retrieval or do not make the information publicly available as they consider them as a company secret. Thus, most cellular network modems still use *AT commands* for configuration and information retrieval. This protocol was introduced in 1981 and reduces the need proprietary drivers to use and control a modem. The manufacturers enhanced the protocol by adding more commands to it, but most manufacturers do not open these extensions to the public.

A collaboration with a cellular modem vendor could further enhance the quality of cellular network measurements with the RMF if more information on the cellular connection was available to the RMF. For example knowing the exact timestamp of the last received resource block containing payload of our measurement traffic could greatly enhance the effectiveness of our *inter-chirp self-induced congestion* (ICSIC) detection algorithm. Additionally, knowledge of the currently used FEC and the spreading factor would give a much tighter upper bound for the currently available data rate. Furthermore, faster information on failing cellular network connectivity could be used — not only by the RMF, but also by the operating systems — to re-initiate the connection faster.

7.2.3 Community driven Cellular Network Traces Database

Evaluating the performance and impact of a given V2X application with our measurement and simulation tool chain presented in this thesis requires sound network traces for the desired simulation area. This requirement currently limits the simulation area to regions personally traceable by the user.

As a consequence the question of the general feasibility of a V2X application using trace-based simulation is only valid for those simulation areas. While simulations can never completely replace real-world FOTs it is desirable to test the V2X application with multiple scenarios and in many different simulation areas.

Thus we suggest to use a community-driven database for high-accuracy cellular network traces gained using the RMF in [1]. This would allow to conduct simulations for areas far away from the person running the simulation.

7.2.4 Measurements with Cell Phones

We currently gather our cellular network traces with standard x86 hardware using our RMF. While this approach allows everyone to build a usable measurement system, it still requires dedicated hardware. In his bachelor thesis [53], Malte Olfen under the supervision of Norbert Goebel ports a previous version of the RMF to android using the android *Native Development Kit (NDK)*. Using this RMF port, we were able to show that current android smartphones easily offer enough calculation power to conduct measurements with the RMF.

But one problem using android smartphones remains, the RMF requires high-precision time synchronization on both the mobile device and the measurement server for the one-way delay measurements and the ICSIC detector. While current smartphones offer access to at least one global positioning system, we did not discover a single android smartphone which offers a pulse-per-second signal and thus allows sub-millisecond time synchronization against a GPS clock as our dedicated hardware does. As stated in our paper [2], time synchronization over the cellular network is no suitable substitution as cellular network communication for the uplink and downlink are not symmetric, but symmetric delays are a requirement for high-accuracy time synchronizations using ntp.

It would be much easier to convince a larger number of users to conduct cellular network traces (like *opsensignal*[1] already does) if a smartphone version of the RMF existed. Thus, as soon as high-accuracy time synchronization via GPS is possible on smartphones, porting a current version of the RMF to android and publishing the app in appstores in combination with a community-driven database (see Section 7.2.3) for

[1] `https://opensignal.com`

high accuracy cellular network traces could help to build up a large simulation basis for trace-based cellular network simulations.

7.3 Closing Words

The automotive industry has vastly changed the face of the world in the last century. Automobiles are affordable for most citizens of industrial nations and thus personal mobility increased dramatically. Furthermore, the industrialization and globalization with just-in-time deliveries require more transports on the roads than ever before. This leads to road networks that can hardly cope with the traffic densities anymore — and even experienced drivers can get overstrained. As a consequence, the number of traffic jams and traffic accidents continuously rises. Inter-vehicular communication with its ability to enhance the knowledge about the surrounding of each connected vehicle beyond its line of sight, promises to raise traffic safety and efficiency by supporting the drivers with detailed information. Furthermore, the driver assistance systems can profit from the enhanced information base gained by IVC. We therefore hope that our work presented in this thesis helps to further enhance traffic safety and efficiency.

Aside from the research presented in this thesis, other factors than IVC exist that have great influence on traffic densities and our environment. Questions arising, but not answered in this thesis, are for example: Is it really necessary that a growing number of employees needs to commute large distances on a daily basis? Are just-in-time production and next-day or even same-day delivery services really needed on a regular basis? Or should we all take on responsibility and switch to public transportation or bicycles as often as possible?

Reducing traffic would also reduce pollution and CO_2 emissions and help to slow down global warming, one of the most imminent challenges of our age.

Hence, "start with the man in the mirror" and "make that change" [54].

Bibliography

Own Publications

[1] N. Goebel, M. Koegel, M. Mauve, and K. Graffi, "Trace-based Simulation of C2X-Communication using Cellular Networks", in *Proceedings of the* 11^{th} *Annual Conference on Wireless On-demand Network Systems and Services (WONS 2014) - Special Session on VANETs and ITS*, Obergurgl, Austria, Apr. 2014, pp. 108–115. DOI: `10.1109/WONS.2014.6814730` (Pages: 5–7, 25–27, 31, 32, 34–36, 52, 53, 62).

[2] N. Goebel, T. Krauthoff, K. Graffi, and M. Mauve, "Moving measurements: Measuring network characteristics of mobile cellular networks on the move ", *Computer Communications*, vol. 97, pp. 81–95, 2017, ISSN: 0140-3664. DOI: `10.1016/j.comcom.2016.06.002` (Pages: 6, 11, 31, 32, 34, 37, 38, 52–54, 62).

[3] N. Goebel, A. Skuballa, M. Mauve, and K. Graffi, "Aggregation and Map-Matching of Mobile Cellular Network Traces", Computer Science Department, Heinrich Heine University, Düsseldorf, Germany, Technical Report: TR-2016-001, Jul. 2016. [Online]: `http://www.cn.hhu.de/nc/publikationen.html?publication=Goebel2016c` (Pages: 6, 36, 37, 52–54).

[4] N. Goebel, R. Bialon, M. Mauve, and K. Graffi, "Coupled Simulation of Mobile Cellular Networks, Road Traffic and V2X applications using Traces", in *Proceedings of the IEEE International Conference on Communications - Mobile and Wireless Networking Symposium (ICC'16 MWN)*, Kuala Lumpur, Malaysia, May 2016, pp. 2875–2881. DOI: `10.1109/ICC.2016.7511126` (Pages: 7, 36–38, 51, 60).

Other References

[5] Statistisches Bundesamt, *Verkehrsunfälle - Fachserie 8 Reihe 7 - Juli 2016*, [Accessed Nov. 1st 2017]. [Online]: https://www.destatis.de/DE/Publikationen/Thematisch/TransportVerkehr/AlteAusgaben/VerkehrsunfaelleMAlt.html (Page: 1).

[6] simTD-Konsortium, *simTD: Sichere Intelligente Mobilität Testfeld Deutschland*, [Accessed Dec. 15th 2017]. [Online]: http://www.simtd.de (Page: 2).

[7] DRIVE C2X consortium, *DRIVE C2X - Accelerate cooperative mobility*, [Accessed Dec. 15th 2017]. [Online]: http://www.drive-c2x.eu/project (Page: 2).

[8] OpenStreetMap, *OpenStreetMap*, [Accessed Dec. 15th 2017], Copyright and License: http://www.openstreetmap.org/copyright. [Online]: http://www.openstreetmap.org/ (Pages: 7, 38).

[9] *The Network Simulator - ns-2*, [Accessed Dec. 16th 2017]. [Online]: http://nsnam.sourceforge.net/wiki (Page: 12).

[10] Ericsson Telecommunicatie B.V., *e.u.r.a.n.e - Enhanced UMTS Radio Access Network Extensions for ns-2*, [Accessed Nov. 19th 2017] (Pages: 12, 13).

[11] M. Vranješ, T. Švedek, and S. Rimac-Drlje, “The use of NS-2 simulator in studying UMTS performances”, *International Journal of Electrical and Computer Engineering Systems*, vol. 1, no. 2, pp. 63–71, 2010 (Page: 12).

[12] A. M. Mutairi and U. A. Baroudi, “NS-2 Enhancements for Detailed HSDPA Simulations”, in *Proceedings of the 3rd International Conference on Performance Evaluation Methodologies and Tools*, ser. ValueTools ’08, Athens, Greece: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, 72:1–72:7, ISBN: 978-963-9799-31-8. DOI: 10.4108/ICST.VALUETOOLS2008.4338 (Page: 13).

[13] G. Piro, N. Baldo, and M. Miozzo, "An LTE module for the ns-3 network simulator", in *SimuTools*, J. Liu, F. Quaglia, S. Eidenbenz, and S. Gilmore, Eds., ICST/ACM, 2011, pp. 415–422, ISBN: 978-1-936968-00-8. DOI: 10.4108/icst.simutools.2011.245571. [Online]: https://dl.acm.org/citation.cfm?id=2151129 (Page: 13).

[14] *LTE-EPC Network simulAtor (LENA) project homepage*, [Accessed Dec. 16th 2017]. [Online]: http://networks.cttc.es/mobile-networks/software-tools/lena (Pages: 13, 14).

[15] *INET Framework - An open-source OMNeT++ model suite for wired, wireless and mobile networks*, [Accessed Dec. 16th 2017]. [Online]: https://inet.omnetpp.org (Page: 15).

[16] A. Virdis, G. Stea, and G. Nardini, "Simulating LTE/LTE-Advanced Networks with SimuLTE", in *Simulation and Modeling Methodologies, Technologies and Applications : International Conference, SIMULTECH 2014 Vienna, Austria, August 28-30, 2014 Revised Selected Papers*, M. S. Obaidat, T. Ören, J. Kacprzyk, and J. Filipe, Eds. Cham: Springer International Publishing, 2015, pp. 83–105, ISBN: 978-3-319-26470-7. DOI: 10.1007/978-3-319-26470-7_5 (Page: 15).

[17] A. Virdis, G. Nardini, and G. Stea, "Modeling unicast device-to-device communications with simuLTE", in *Proceedings of the 1st International Workshop on Link- and System Level Simulations (IWSLS)*, Jul. 2016, pp. 1–6. DOI: 10.1109/IWSLS.2016.7801579 (Page: 15).

[18] M. Wrulich, W. Weiler, and M. Rupp, "HSDPA Performance in a Mixed Traffic Network", in *Proceedings of the VTC Spring 2008 - IEEE Vehicular Technology Conference*, May 2008, pp. 2056–2060. DOI: 10.1109/VETECS.2008.462 (Pages: 16, 17).

[19] C. Mehlführer, M. Wrulich, J. C. Ikuno, D. Bosanska, and M. Rupp, "Simulating the Long Term Evolution physical layer", in *Proceedings of the 17th European Signal Processing Conference*, IEEE, Aug. 2009, pp. 1471–1478 (Pages: 16, 17).

[20] Mathworks, *Parallel Computing Toolbox homepage*, [Accessed Dec. 16th 2017]. [Online]: https://www.mathworks.com/products/parallel-computing.html (Page: 16).

[21] G. Piro, L. A. Grieco, G. Boggia, F. Capozzi, and P. Camarda, "Simulating LTE Cellular Systems: An Open-Source Framework", *IEEE Transactions on Vehic-*

ular Technology, vol. 60, no. 2, pp. 498–513, Feb. 2011, ISSN: 0018-9545. DOI: 10.1109/TVT.2010.2091660 (Page: 17).

[22] R. Protzmann, B. Schünemann, and I. Radusch, "Simulation of Convergent Networks for Intelligent Transport Systems with VSimRTI", in *Networking Simulation for Intelligent Transportation Systems*. John Wiley & Sons, Inc., 2017, pp. 1–28, ISBN: 9781119407447. DOI: 10.1002/9781119407447.ch1 (Page: 18).

[23] C. Sommer, J. Härri, F. Hrizi, B. Schünemann, and F. Dressler, "Simulation Tools and Techniques for Vehicular Communications and Applications", in *Vehicular ad hoc Networks: Standards, Solutions, and Research*, C. Campolo, A. Molinaro, and R. Scopigno, Eds. Cham: Springer International Publishing, 2015, pp. 365–392, ISBN: 978-3-319-15497-8. DOI: 10.1007/978-3-319-15497-8_13 (Page: 20).

[24] *MiXiM (mixed simulator) on github*, [Accessed Dec. 16th 2017]. [Online]: http://github.com/omnetpp/mixim (Page: 20).

[25] A. Köpke, M. Swigulski, K. Wessel, D. Willkomm, P. T. K. Haneveld, T. E. V. Parker, O. W. Visser, H. S. Lichte, and S. Valentin, "Simulating Wireless and Mobile Networks in OMNeT++ the MiXiM Vision", in *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops*, ser. Simutools '08, Marseille, France: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, 71:1–71:8, ISBN: 978-963-9799-20-2 (Page: 20).

[26] *Mobilty Framework for OMNeT++*, [Accessed Dec. 15th 2017]. [Online]: http://mobility-fw.sourceforge.net (Page: 20).

[27] C. Sommer, R. German, and F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis", *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, Jan. 2011. DOI: 10.1109/TMC.2010.133 (Page: 21).

[28] A. Wegener, M. Piórkowski, M. Raya, H. Hellbrück, S. Fischer, and J.-P. Hubaux, "TraCI: An Interface for Coupling Road Traffic and Network Simulators", in *Proceedings of the 11^{th} Communications and Networking Simulation Symposium*, ser. CNS '08, Ottawa, Canada: ACM, Apr. 2008, pp. 155–163, ISBN: 1-56555-318-7. DOI: 10.1145/1400713.1400740 (Page: 21).

[29] *Vehicles in Network Simulation (Veins) homepage*, [Accessed Dec. 16th 2017]. [Online]: http://veins.car2x.org (Page: 21).

[30] F. Hagenauer, F. Dressler, and C. Sommer, "Poster: A simulator for heterogeneous vehicular networks", in *Proceedings of the IEEE Vehicular Networking Conference (VNC)*, Dec. 2014, pp. 185–186. DOI: 10.1109/VNC.2014.7013339 (Page: 21).

[31] *Veins LTE homepage*, [Accessed Dec. 16th 2017]. [Online]: http://veins-lte.car2x.org (Page: 21).

[32] *iTETRIS platform homepage*, [Accessed Dec. 16th 2017]. [Online]: http://ict-itetris.eu (Page: 21).

[33] M. Rondinone, J. Maneros, D. Krajzewicz, R. Bauza, P. Cataldi, F. Hrizi, J. Gozalvez, V. Kumar, M. Röckl, L. Lin, O. Lazaro, J. Leguay, J. Härri, S. Vaz, Y. Lopez, M. Sepulcre, M. Wetterwald, R. Blokpoel, and F. Cartolano, "iTETRIS: A modular simulation platform for the large scale evaluation of cooperative ITS applications", *Simulation Modelling Practice and Theory*, vol. 34, no. Supplement C, pp. 99–125, 2013, ISSN: 1569-190X. DOI: 10.1016/j.simpat.2013.01.007 (Page: 21).

[34] *Homepage for the Improvements to the University of Strathclyde ns-2 UMTS module*, [Accessed Dec. 16th 2017]. [Online]: https://staff.info.unamur.be/lsc/Research/ns-2_UMTS (Page: 22).

[35] B. Schünemann, "V2X simulation runtime infrastructure VSimRTI: An assessment tool to design smart traffic management systems", *Computer Networks*, vol. 55, no. 14, pp. 3189–3198, 2011, ISSN: 1389-1286. DOI: 10.1016/j.comnet.2011.05.005 (Page: 22).

[36] Daimler Center for Automotive IT Innovations (DCAITI), *VSimRTI homepage*, [Accessed Dec. 16th 2017]. [Online]: https://www.dcaiti.tu-berlin.de/research/simulation (Page: 22).

[37] TETCOS, *NetSim homepage*, [Accessed Dec. 16th 2017]. [Online]: http://www.tetcos.com/index.html (Page: 22).

[38] ——, *NetSim Brochure*, [Downloaded May. 28th 2017]. [Online]: http://www.tetcos.com/downloads/NetSim_Brochure.pdf (Page: 22).

[39] J. Strauss, D. Katabi, and F. Kaashoek, "A Measurement Study of Available Bandwidth Estimation Tools", in *Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '03, Miami Beach, FL, USA: ACM, 2003, pp. 39–44, ISBN: 1-58113-773-7. DOI: 10.1145/948205.948211 (Page: 33).

[40] N. Hu and P. Steenkiste, "Evaluation and characterization of available bandwidth probing techniques", *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 6, pp. 879–894, 2003. DOI: 10.1109/JSAC.2003.814505 (Page: 33).

[41] V. Ribeiro, M. Coates, R. Riedi, S. Sarvotham, B. Hendricks, and R. Baraniuk, "Multifractal cross-traffic estimation", in *Proceeding of the ITC Specialist Seminar on IP Traffic Measurement, Modeling, and Management*, 2000, pp. 15–1. [Online]: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.29.9411 (Page: 33).

[42] B. Melander, M. Bjorkman, and P. Gunningberg, "A new end-to-end probing and analysis method for estimating bandwidth bottlenecks", in *Proceedings of the Global Telecommunications Conference, 2000. GLOBECOM '00. IEEE*, vol. 1, 2000, pp. 415–420. DOI: 10.1109/GLOCOM.2000.892039 (Page: 33).

[43] M. Jain and C. Dovrolis, "Pathload: A Measurement Tool for End-to-End Available Bandwidth", in *Proceedings of the Passive and Active Measurements (PAM) Workshop*, 2002, pp. 14–25. [Online]: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.17.4495 (Page: 33).

[44] V. J. Ribeiro, R. H. Riedi, R. G. Baraniuk, J. Navratil, and L. Cottrell, "PathChirp: Efficient Available Bandwidth Estimation for Network Paths", in *Proceedings of the Passive and Active Measurement (PAM) Workshop*, 2003. DOI: 10.2172/813038 (Page: 33).

[45] E. Goldoni, G. Rossi, and A. Torelli, "Assolo, a New Method for Available Bandwidth Estimation", in *Proceedings of the 4^{th} International Conference on Internet Monitoring and Protection*, May 2009, pp. 130–136. DOI: 10.1109/ICIMP.2009.28 (Page: 33).

[46] *iperf2 - Incremental enhancements to iperf*, [Accessed Dec. 15^{th} 2017]. [Online]: http://sourceforge.net/projects/iperf2/ (Page: 33).

[47] *iperf3: A TCP, UDP, and SCTP network bandwidth measurement tool*, [Accessed Dec. 15^{th} 2017]. [Online]: https://github.com/esnet/iperf (Page: 33).

[48] *speedtest.net / Ookla*, [Accessed Dec. 15^{th} 2017]. [Online]: http://speedtest.net (Page: 33).

[49] *speedtest.net measurement explained*, [Accessed Dec. 15^{th} 2017]. [Online]: https://support.speedtest.net/hc/en-us/articles/203845400-How-does-the-test-itself-work-How-is-the-result-calculated- (Page: 33).

[50] F. Marchal, J. Hackney, and K. W. Axhausen, "Efficient Map-Matching of Large GPS Data Sets - Tests on a Speed Monitoring Experiment in Zurich, volume 244 of Arbeitsbericht Verkehrs- und Raumplanung", Technical Report: 2004. [Online]: `http://citeseerx.ist.psu.edu/viewdoc/citations?doi=10.1.1.113.4605` (Pages: 38, 39).

[51] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "SUMO – Simulation of Urban MObility", in *Proceedings of the* 3^{rd} *International Conference on Advances in System Simulation (SIMUL), Barcelona, Spain*, 2011, ISBN: 978-1-61208-169-4. [Online]: `http://elib.dlr.de/71460/` (Page: 38).

[52] A. Varga *et al.*, "The OMNeT++ discrete event simulation system", in *Proceedings of the European simulation multiconference (ESM'2001)*, vol. 9, 2001, p. 65 (Pages: 52, 54).

[53] M. Olfen, *Ende-zu-Ende-Messungen von Mobilfunkparametern mit Smartphones*, Bachelor's thesis, Jun. 2013. [Online]: `http://www.cn.hhu.de/publikationen.html?publication=Olfen2013a` (Page: 62).

[54] M. Jackson, *Man in the Mirror* (Page: 63).

Eidesstattliche Erklärung
laut §5 der Promotionsordnung vom 06.12.2013
der Mathematisch-Naturwissenschaftlichen Fakultät
der Heinrich-Heine-Universität Düsseldorf

Ich versichere an Eides Statt, dass die Dissertation von mir selbständig und ohne unzulässige fremde Hilfe unter Beachtung der „Grundsätze zur Sicherung guter wissenschaftlicher Praxis an der Heinrich-Heine-Universität Düsseldorf“ erstellt worden ist.

Ort, Datum	Norbert Goebel