TECHNISCHE
UNIVERSITÄT
DARMSTADT

DYNAMIC MAPS *for* HIGHLY AUTOMATED DRIVING

Generation, Distribution and Provision

Dem Fachbereich Elektrotechnik und Informationstechnik
der Technischen Universität Darmstadt
zur Erlangung des akademischen Grades eines
Doktor-Ingenieurs (Dr.-Ing.)
vorgelegte Dissertation

von

FLORIAN JOMRICH, M.SC.

Geboren am 17. Januar 1990 in Rüsselsheim am Main

Referent: Prof. Dr.-Ing. Ralf Steinmetz
Korreferent: Prof. Dr.-Ing. Ioannis Stavrakakis

Tag der Einreichung: 06. August 2019
Tag der Disputation: 02. Dezember 2019

Hochschulkennziffer D17
Darmstadt 2019

Florian Jomrich, M.Sc.: *Dynamic Maps for Highly Automated Driving,*
Generation, Distribution and Provision
Darmstadt, Technische Universität Darmstadt,
Jahr der Veröffentlichung der Dissertation auf TUprints: 2020
Tag der mündlichen Prüfung: 02.12.2019

Dieses Dokument wird bereitgestellt von: This document is provided by:
tuprints, E-Publishing-Service der Technischen Universität Darmstadt.

http://tuprints.ulb.tu-darmstadt.de
tuprints@ulb.tu-darmstadt.de

Bitte zitieren Sie dieses Dokument als: Please cite this document as:

URN: urn:nbn:de:tuda-tuprints-97022
URL: http://tuprints.ulb.tu-darmstadt.de/id/eprint/9702

Die Veröffentlichung steht unter folgender Creative Commons Lizenz:
Namensnennung - Nicht kommerziell - Keine Bearbeitungen 4.0 International
CC BY-NC-ND 4.0
https://creativecommons.org/licenses/by-nc-nd/4.0/deed.de

This publication is licensed under the following Creative Commons License:
Attribution - Non Commercial - No Derivatives 4.0 International
CC BY-NC-ND 4.0
https://creativecommons.org/licenses/by-nc-nd/4.0/deed.de

ACKNOWLEDGEMENTS

I say thank you to all the people, who supported me to complete this work. First of all thanks to Professor Ralf Steinmetz and Professor Ioannis Stavrakakis for supervising me and all their valuable provided feedback towards my work. Furthermore I thank Doreen Böhnstedt, Björn Richerzhagen, Steffen Knapp and Harald Berninger for their helpful feedback during my research activities. Thank you to all my colleagues at the Multimedia Communication Department of the Technical University Darmstadt, especially the members of my research group "Distributed Sensing Systems". Furthermore I greatly thank all my colleagues at the Opel plant in Rüsselsheim. Big thanks also go to my family, especially my mom, and all my friends who helped me in pursuing the task to write this thesis. I say thank you for an exciting time in my life and the knowledge I obtained from it. I hope when reading this work one remembers the conversations we had and the time we spent together.

ABSTRACT

With an ever-increasing number of vehicles roaming the streets and a general intensification of ongoing daily traffic the current vehicular safety systems are not able to reduce the number of traffic accidents further.

As the majority of severe or deadly traffic accidents nowadays is caused by human error, car manufacturers and researchers alike focus on the self-driving vehicle as a promising solution to this problem, as a machine is unaffected from human conditions such as tiredness or drunk driving.

To enhance the overall achievable driving safety and comfort the self-driving vehicles rely on an additional *map database*, besides the hardware sensor system installed onboard.

The so-called *High Definition Map (HD Map)*, a highly precise virtual model of the actual real-world provides detailed information about the ongoing traffic situation ahead of the car's sensor ranges. Otherwise critical traffic situations can be resolved by this a priori knowledge and if necessary, a handover of the driving control back to a human driver can be triggered.

The *maintenance* of the HD Map is a major challenge, as due to the importance of the map for the self-driving vehicle map updates have to be realized in much shorter time (minutes instead of months) compared to established concepts common for human-oriented digital navigation maps.

This thesis provides contributions in the areas of *Distribution, Generation and Provision* of such map updates, as the key communication challenges of the maintenance procedure.

Our first contribution is the development, implementation and evaluation of a protocol that realizes the *context-specific distribution* of partial and incremental map updates. The protocol has been designed with the prerequisites and requirements of a self-driving vehicle in mind. To achieve the efficient dissemination of updates to all cars the protocol relies on infrastructure-based (cellular) and ad hoc communication (WLAN) between the vehicles. The performance of the protocol is evaluated based on realistic traffic simulations and actual map content.

As our second contribution, we develop and implement an algorithm that *detects changes* in the road infrastructure (e.g. induced by construction sides) based solely on low-cost sensor information. This detection algorithm facilitates the succeeding update generation of the map data in the identified area. We evaluate the capabilities of the detection algorithm under a real-world data set in the example of a highway construction site scenario.

To enhance the *provision* of map updates and vehicular sensor data via wireless communication, we conduct our third and most comprehensive contribution. We focus on the design and enhancement of a variety of different techniques and concepts to obtain broad knowledge about the serving wireless network to be provided in a sub-

sequent step as valuable information to related transmission scheduling algorithms. These techniques and concepts include the measurement and prediction of the various performance indicators of actual deployed cellular networks, via low-cost hardware and software, as well as their further usage in simulation and network connectivity maps, always with an emphasis on easy deployability and the reutilization of existing components.

Overall, this thesis presents essential contributions, which in their collectivity support the realization of a robust, dynamic and reliable maintenance cycle of an HD Map for self-driving vehicles.

KURZFASSUNG

Durch die ständig anwachsende Zahl an Fahrzeugen auf den Straßen und die allgemeine Intensivierung des täglichen Verkehrs sind die aktuell verfügbaren Fahrzeugsicherheitssysteme nicht mehr dazu in der Lage die Zahl der auftretenden Verkehrsunfälle weiter zu reduzieren.

Da die meisten schweren oder tödlichen Verkehrsunfälle heute durch menschliches Versagen verursacht werden, forschen Automobilhersteller und Wissenschaftler gleichermaßen an selbstfahrenden Fahrzeugen als vielversprechende Lösung für dieses Problem, da eine Maschine nicht von menschlichen Eigenschaften wie z.B. Müdigkeit oder Alkoholisierung beeinflusst wird.

Um die insgesamt erreichbare Fahrsicherheit und den Fahrkomfort der selbstfahrenden Fahrzeuge zu gewährleisten, kommt neben den an Bord installierten Sensorsystemen eine zusätzliche Kartendatenbank zum Einsatz.

Die so genannte *High Definition Map (HD Karte)*, ein hochpräzises virtuelles Modell der realen Welt liefert detaillierte Informationen über die aktuelle Verkehrssituation außerhalb des Blickfeldes des Fahrzeugs. Mit diesem a priori Wissen können andernfalls kritische Verkehrssituationen umgangen oder falls notwendig eine komfortable Übergabe der Fahrsteuerung an einen menschlichen Fahrer ausgelöst werden.

Die *Wartung* der HD Karte ist ein offenes Forschungsfeld, da aufgrund der Bedeutung der Karte für die selbstfahrenden Fahrzeugkarte Updates in wesentlich kürzerer Zeit (Minuten statt Monaten) realisiert werden müssen, als dies die etablierten Konzepte für digitale Navigationskarten von menschlichen Fahrern im Stande zu leisten sind.

Die vorliegende Arbeit liefert hierzu wesentliche Beiträge in den Bereichen *Generierung, Verteilung und Bereitstellung* solcher Kartenupdates, als die zentralen kommunikative Herausforderungen in Bezug auf die Kartenwartung.

Unser erster Beitrag in diesem Zusammenhang ist die Entwicklung, Implementierung und Evaluierung eines Protokolls, das die *kontextspezifische Verbreitung* von partiellen und inkrementellen Kartenupdates realisiert. Das Protokoll wurde dabei unter Berücksichtigung der spezifischen Voraussetzungen und Anforderungen eines selbstfahrenden Fahrzeugs entwickelt. Um eine effiziente Verbreitung von Updates unter allen betroffenen Fahrzeugen zu realisieren greift es dabei wahlweise auf infrastrukturgebundene (zelluläre) oder ad hoc basierte (WLAN), drahtlose Kommunikation zurück. Die Performanz des Protokolls wird anhand realistischer Verkehrssimulationen und realer Karteninhalte bewertet.

Als zweiten Beitrag entwickeln und implementieren wir einen Algorithmus, der Veränderungen in der Straßeninfrastruktur (z.B. verursacht durch Baustellen) erkennt, der ausschließlich auf kostengünstigen Sensorinformationen basiert. Der Erkennungsalgorithmus ermöglicht damit eine anschließende schnelle, da spezifische Aktualisierung

des Kartenmaterials im identifizierten Bereich. Die Performanz des Algorithmus wird am realen Beispiel einer Verkehrsbaustelle auf der Autobahn bewertet.

Um die Bereitstellung der genannten Kartenupdates und Fahrzeugsensordaten durch drahtlose Kommunikation zu verbessern, leisten wir unseren dritten und umfassendsten Beitrag. Dabei konzentrieren wir uns auf das Design und die Verbesserung vielfältiger Techniken und Konzepte, um ein breites Wissen über das zugrundeliegende Netzwerk zur drahtlosen Kommunikation zu erhalten. Dieses wertvolle Wissen wird dann in einem späteren Schritt bereits existierenden Algorithmen zur besseren Planung der Datenübertragung bereit gestellt. Zu diesem Beitrag zählen die Messung und Vorhersage einer Auswahl der verschiedenen Leistungsindikatoren der derzeit (August 2019) eingesetzten Mobilfunknetze über kostengünstige Hard- und Software, sowie deren Weiterverwendung in der Simulation und der Generierung von sogenannten Verbindungskarten der Netzqualität. Der Fokus unserer Beiträge liegt hierbei auf der einfachen, großflächigen Anwendung unter der Nutzung bereits bestehender Komponenten.

Zusammenfassend präsentiert diese Arbeit zentrale Beiträge, die in ihrer Gesamtheit die Realisierung einer robusten, dynamischen und zuverlässigen Wartung einer HD Karte für selbstfahrende Fahrzeuge wesentlich unterstützen.

CONTENTS

1 INTRODUCTION 1
1.1 Motivation for Dynamic Maps for Highly Automated Driving 2
1.2 Research Challenges 5
1.3 Research Goals and Contributions 7
1.4 Structure of the Thesis 9

2 BACKGROUND 11
2.1 Specification of Automation Levels 11
2.2 HD Map and Related Advanced Driver Assistance Systems 12
2.3 Importance of the Cellular Network for Self-Driving Vehicles 15
2.4 Position Dilution of Precision of Global Navigation Satellite Systems . . 16
2.5 Geographic Location via Geohashes 17
2.6 Simulation of Urban Mobility - Traffic Simulator - SUMO 18

3 RELATED WORK 21
3.1 Map Updates for Digital Navigation Systems 21
3.2 Vehicular Sensor Data Aggregation to Maintain Map Data 25
3.3 Optimized Provision Using Infrastructure-Managed Cellular Networks 29
3.4 Summary and Identified Research Gaps 33

4 MAP UPDATES FOR HIGHLY AUTOMATED VEHICLES 37
4.1 Dynamic Map Update Protocol 37
4.2 HD-Wmap Extension 48

5 ROAD INFRASTRUCTURE CHANGE DETECTION 57
5.1 Design of the Lane Course Change Detection Algorithm 57
5.2 Creation of Test Data Set 65
5.3 Evaluation of Clustering Performance 67
5.4 Evaluation of Change Detection - Highway Construction Site Scenario . 76

6 OPTIMIZED DATA PROVISION FOR HIGHLY AUTOMATED DRIVING 81
6.1 Communication Requirements of Highly Automated Vehicles 82
6.2 Measurement Campaign Conclusions - Connectivity Map as HD Map Layer 97
6.3 Connectivity Map Creation Framework - ICCOMQS 97
6.4 Connectivity Map Supported Online Throughput Estimation 113
6.5 Mobile Estimation of Active Clients Using LTE Control Channel Decoding 127
6.6 Further Usage of Measurement Data in Simulation 129

7 SUMMARY, CONCLUSIONS, AND OUTLOOK 141
7.1 Summary of the Thesis 141
7.2 Outlook 144

BIBLIOGRAPHY 147

A APPENDIX 169
A.1 Definition of Automation Levels 169
A.2 Prototypical Implementation of an Operational HD Map - Example of the Ko-HAF Project 171
A.3 Key Performance Indicators of the Cellular Network 173
A.4 Map Changes in Digital Navigation Map Data of HERE 176
A.5 Network Coverage Maps of Cellular Providers 178
A.6 Detailed Listing of the Evaluation Results of the HD-Wmap Extension of the Dynamic Map Update Protocol 179
A.7 Right Lane Evaluation Results of the Lane Course Detection Algorithm 180
A.8 Detailed Listing of the Cellular Network Measurements Conducted During the Ko-HAF Project 182
A.9 Detailed Calculation for Frankfurt-Niederrad Scenario 184
A.10 Evaluation of Opel Proving Ground in Rodgau-Dudenhofen 186
A.11 Simulated SUMO Scenarios 194
A.12 Detailed List of Cellular Network Quality Parameters Collected Through the Custom Android Measuring Application 197
A.13 Localized vs Globalized Training Based on MobileInsight Data from Provider B and C 201
A.14 Detailed Correlation Plots for the Online Estimation 204
A.15 Mobile Test of Imdea OWL Connected to ConnectivityMap Client in Vehicle 210
A.16 List of Acronyms 215
A.17 Supervised Student Theses 217

B AUTHOR'S PUBLICATIONS 219

C CURRICULUM VITÆ 223

D ERKLÄRUNG LAUT PROMOTIONSORDNUNG 225

1

INTRODUCTION

Continuous development of active and passive safety systems, for example ESP, ABS, the airbag or the brake assist, has tremendously increased the safety of motoring since the beginning of the car manufacturing.

Motivation for self-driving vehicles

However with a forecast of continuous increase of traffic in the near future [1, 2], the number of severe and deadly traffic accidents is expected to rise again despite the currently available safety features at hand. The reasons for this increase are manifold like continuous urbanization and growth in city sizes, as well as the further aging of the population and technical advancements such as smartphones [3].

These and other related trends lead car manufacturers and researchers to think about new mechanisms on how driving safety can be further increased and how nowadays drivers can be supported.

88% of all German traffic accidents were related to human driver's fault in the year 2017 [4]. So self-driving and -communicating cars are seen as a major opportunity to increase driving safety. Autonomously driving vehicles are not influenced by human errors such as drunk or overtired driving. The idea for self-driving cars is nearly as old as the car itself dating back to at least the first decades of the 20th century [5]. Besides the overall improvement of safety and driving efficiency, the increase of comfort while traveling is also seen as an important selling point for future customers [6].

Vehicle sensor setup

Through several major recent advances in many different technological areas, such as sensing systems and computational power, the self-driving vehicle is now about to become a reality [7]. The autonomous car can rely on a multitude of sensors, for example radar (radio detection and ranging), ultra sonic sensors, cameras, lidar (light detection and ranging) as well as Global Navigation Satellite Systems (GNSS) [8] and their intelligent fusion. As a result, it can predict and avoid critical driving situations, while achieving reaction times and viewing capabilities that go far beyond those, any human could possibly achieve.

HD Map as virtual sensor

However with all its sensing capabilities the self-driving vehicle still might experience dangerous or uncomfortable driving situations, that can only be properly avoided with a foresight, that goes beyond the own sensor range of the car. A traffic jam or an accident behind a corner or ice on the road ahead are just some of such examples. To address such situations and to ensure the human safety and trust in the technical systems, the self-driving vehicles also rely on an additional "virtual" sensor for their driving task, the so called High Definition Map (HD Map) [9–12]. This map provides detailed and critical information to the self-driving vehicles regarding their current surrounding traffic situation and environment. The provided information is highly accurate geo-referenced up to the sub-meter level of precision. In comparison nowadays, standard navigation map data only achieves a localization accuracy of several meters. The HD Map thus can be seen much more as a highly precise virtual 3D rep-

representation of the actual real world [10, 13]. That way the self-driving car can compare its own sensor-readings with a virtual reference, which facilitates the driving task significantly.

Maintenance challenge

Due to the high degree of precision and the time criticality of the contained information the HD Map must be quickly updated. The general problem already present for normal navigation systems map data [14] is amplified by several magnitudes [12]. The self-driving vehicles have to be able to always rely on the most up to date map data to properly execute driving decisions while on the road. As result the HD Map has to be continuously maintained and updates must be delivered wirelessly to the self-driving car as indicated in Figure 1. To obtain information regarding the current traffic situation from public authorities, such as the police, is not sufficient enough therefore. Also the self-driving vehicles themselves are required to continuously collect information about the current traffic situation. Without their sensor data being uploaded to a data processing and aggregating entity, the sufficient generation of necessary map updates might not be achieved. To develop and assess new methodologies to maintain this data loop is the key contribution of this thesis as explained in the following.

1.1 MOTIVATION FOR DYNAMIC MAPS FOR HIGHLY AUTOMATED DRIVING

By relying on the HD Map the self-driving car increases its capabilities to gather intel about the current traffic situation around itself and ahead along its track, beyond the area read by its own sensor's range and capabilities. Besides safety relevant information for the strategic planning of the driving task, the HD Map enhances the other on board sensor's performance by providing them with a virtual reference [12, 13, 16–18]. This reference can be used to compare the sensor's readings with an expected value (provided by the HD Map) [11, 18, 19].

HD Map enhances driving comfort and safety

By combing all information the performance of the sensory results is improved. In conclusion this yields a higher degree of comfort and performance under highly automated driving. For example, the exact localization of the vehicle's current position on the street can be improved by fusing different sensor readings (GNSS, wheel ticks, accelerometer, ...) together and combining their information based on the map data. The camera of an automated vehicle for example can be used to identify landmarks that are precisely geo-referenced within the map. These geo-referenced positions are then fused into the car's current position estimation (based on Global Navigation Satellite Systems data) and improve the overall localization accuracy.

Virtual representation of the world

> "Rather than having to figure out what the world looks like and what it means from scratch every time we turn on the (driving AI) software, we tell it what the world is expected to look like when it is empty (provided through the HD Map). And then the job of the software is to figure out how the world is different from that expectation. This makes the problem a lot simpler." (Andrew Chatham, the Google self-driving car team's mapping lead - 2014 [13])

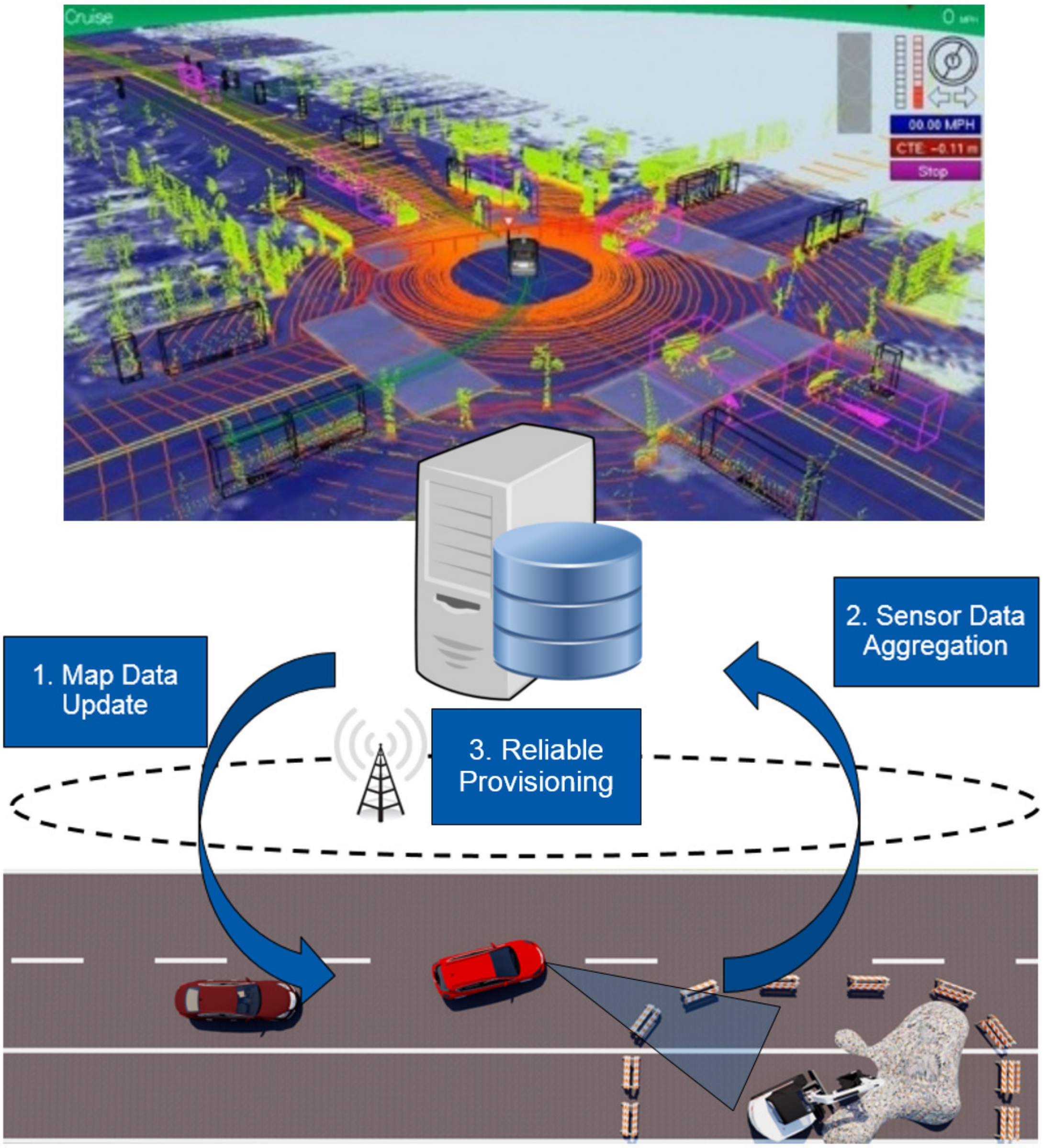


Figure 1: General working principle of a high definition street map for highly automated driving vehicles [15]. (Contains image ©Google[1])

To be able to fully rely on the provided information the self-driving vehicle requires always the most up to date map data and traffic information (illustrated by Figure 1). A continuous loop of data exchange between the self-driving cars and an information processing entity, such as a central backend server, is necessary to maintain the reliability of the HD Map . Thus a robust wireless communication is further a key requirement. The successful operation of the HD Map in the self-driving car addresses similar requirements that have been present for previous digital navigation maps and

---

[1] https://digital.hbs.edu/platform-digit/submission/google-x-leveraging-data-and-algorithms-for-self-driving-cars/ (Last accessed on August 1, 2019)

driver assistance features. However as the HD Maps main functionality is to provide information to a machine (the self-driving vehicle) rather than a human driver, those requirements are taken to a new level of mandatory performance not available by currently existing technologies.

Evolution of digital navigation map maintenance

Assisting the human driver with digital map data and road guidance became possible with the advent of Global Navigation Satellite Systems, such as the American Global Positioning System (GPS) [20]. The first digital navigation system has been introduced in 1991 by Toyota [21]. The initial approach to maintain the consistency and route guidance functionality of the map over time was a necessary full replacement of the binary map data [22–24] after several months. This kind of update procedure is still a common way till today for car navigation systems[2] [25]. With the introduction of mobile communication functionalities inside the vehicles [26] more time efficient and wireless updating mechanisms such as partial and incremental map updates [22, 23, 27, 28] have been developed. These new updating technologies however now had to ensure the consistency and the road guidance functionality of the map [29] after a partial update. The availability of wireless updates also enabled the provisioning of current traffic information to the human driver [26, 30]. Over time these general concepts have been further refined and enhanced resulting in more advanced technologies such as an electronic horizon (e-Horizon) [30–35].

Cars as mobile sensors

To leverage sensor data from static or (as in the chase of a self-driving car) mobile sensors to derive meaningful information about their current environment is a well established research domain. Many different terms [36] have been established to address different aspects of this generalized concept. In the mobile domain, which is most relevant for self-driving vehicles this includes the terms participatory sensing [37], crowdsourcing [38], ubiquitos mapping [39] and floating car data [40]. The swarm intelligence present in nowadays vehicular traffic can be used for many different applications by collecting and processing the vehicles onboard sensors data. This includes the creation or the update of existing navigation map data [41–46] or to derive the current driving behaviour and exact position of the car [47–51].

Wireless communication realizes vehicular services

The fundamental basis to enable the described necessary data exchange and any additional connected services for a self-driving vehicle is a set of reliable and efficient wireless communication technologies. Over the years several different communication interfaces have been tested and deployed into cars, including systems relying on communication infrastructure (Car to Infrastructure - C2I), such as cellular network towers [52, 53] or WLAN-Road Side Units [54–56] and systems enabling direct communication between the vehicles themselves (Car to Car - C2C) [57]. Performance and reliability of the established connections always have been a major criteria to ensure safety [58–61] as well as quality of experience of non-safety critical services for the customer [62]. To support the vehicles in their task of data transmission, several different technologies have been developed to provide them with performance indicators of the current [63–74] and to be expected [75–80] network quality along their trip.

Enhanced requirements for self-driving vehicles

---

[2] https://us.support.tomtom.com/app/answers/minor_detail/a_id/9079/ (Last accessed on August 1, 2019)

The self-driving vehicle, as an important entity of the future massive and secure machine type communication [52, 53, 80], now requires new performance advancements in the afore mentioned technological research areas that had not been required for human drivers before. Therefore, *i)* new methodologies have to be developed to provide more frequent HD Map updates as data-efficient and as reliable as possible to all self-driving vehicles. This includes as well *ii)* a constant verification of the existing map data's correctness based on a multitude of different sensor data providers, like road authorities and the self-driving vehicles themselves. To ensure this data exchange the *iii)* intelligent combination and evaluation of existing communication paradigms and mechanisms specifically for the application requirements of a self-driving car is an additional key aspect, as addressed in this thesis.

1.2 RESEARCH CHALLENGES

To ensure the functionality of the HD Map a continuous data loop of map update downloads and sensor data uploads is established between the self-driving vehicles and a data processing entity (e.g. a backend server in Figure 1). The wireless communication infrastructure as a connecting medium therefore plays another important role. A robust data communication has to be ensured throughout the whole trip of the self-driving car. All these three components inside the data loop, the data-efficient and timely provisioning of the map updates, the sensor data upload, as well as the robust wireless cellular communication, pose new challenges:

Challenge: Reliable and Data-efficient Distribution of Map Data Updates

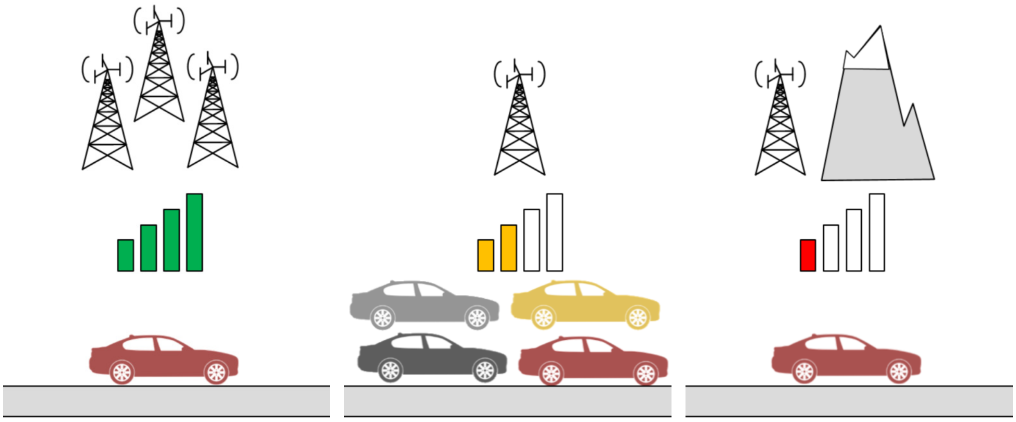
HD Maps contain a largely increased amount of highly accurate information compared to common digital navigation map data of nowadays (2019). Due to this reason the currently established provisioning concepts for digital map data are insufficient for HD Maps in many ways. Common updating cycles of map data of several months [22, 23] have to be completed now in far less time to ensure the functionality of the HD Map [18]. Update cycles of some hours to only minutes become necessary, depending on the criticality of the update. Even advanced distribution concepts such as partial and incremental updates of map data [22–24, 29, 81–83] still poses a huge overhead in costly data transmission, as they have been developed for data required by humans and not self-driving cars in mind. The HD Map's main purpose, to provide guidance for self-driving vehicles, now renders it necessary to think about more specific updating concepts in terms of data efficiency and provisioning methods. A robust wireless communication is a key requirement therefore. Existing wireless technologies, such as cellular and Wi-Fi, all have their advantages and disadvantages in comparison (e.g. transmission-costs, coverage, transmission-speed). To select the most suitable technology requires a constant reconsideration of further influencing factors such as traffic density and the overall destination of the vehicles.

Challenge: Fast Road Infrastructure Change Detection

The reliable operation of the HD Map requires continues updates about the current traffic situation. External entities, such as the police or federal agencies often might require an insufficient amount of time to provide such necessary information. Also the amount of self-driving vehicles in certain map areas might be too spare to successfully rely on their sensor data alone. Low-cost sensors do not provide the same level of sensor accuracy, but are ubiquitous available for example in current production vehicles and aftermarket devices such as smartphones and wearables. These different data sources with varying quality and accuracy require new techniques to combine their information together to obtain a more accurate picture of the current traffic environment. Especially the initial detection of faulty map data is a time-critical and highly safety relevant task necessary to warn affected self-driving vehicles of otherwise dangerous traffic situations. In summary an intelligent aggregation and post processing of as many different data sources as possible is necessary to keep the HD Map on the highest possible level of functionality and performance.

Challenge: Reliable Data Provisioning Via Wireless Infrastructure-Based Communication

Due to the mobility of the self-driving vehicles, their experienced Quality of Service (QoS), when using infrastructure-based network communication, for example the cellular network, is constantly changing. Throughout a single trip, an (e.g. urban) area of good network coverage (Fig. 2a) with high achievable data rates can quickly be replaced by an (e.g. rural) area along the track with only poor network connection or no available network at all (Fig. 2c). The available transmission technologies are influenced by a variety of different factors. This includes static conditions such as the amount of deployed network access points in the surrounding area, but also dynamic and highly dynamic factors such as the amount of currently active users [65, 84] (Fig. 2b) or the current weather [85, 86]. Extreme situations such as crowded traffic during rush hours or heavy rain can have a critical impact on the overall achievable network performance. Additionally data services, like video or music streaming, that are requested by the passengers of the self-driving vehicles, introduce another significant load on the network. In this environment the maintenance of the HD Map requires a reliable scheduling and transmission of map updates and vehicular sensor data. Especially the vehicular sensor information generates significant amounts of data as it has to be continuously uploaded to the processing server to verify the correctness of the most recent HD Map data. For many wireless networks, such as the cellular network, this becomes an even more challenging task, as they provide an asymmetric connection for the vehicles [87]. For such media the achievable data rate in the upload direction is multiple factors smaller than the one in the download direction. In conclusion this makes the maintenance of the HD Map for self-driving cars an especially challenging task in terms of required network performance.



(a) good infrastructure (b) varying number of cellular users (c) environmental conditions

Figure 2: Challenging influence factors in the cellular environment due to the vehicles mobility.

1.3 RESEARCH GOALS AND CONTRIBUTIONS

The main goal of this work is to develop, realize and evaluate new mechanisms that ensure the data-efficient and robust operation of the HD Map inside a self-driving car, as described previously by the data loop in Figure 1. This objective is divided into the following major research goals.

Research Goal 1: Efficient HD Map Update Distribution for Self-Driving Vehicles

To develop new data distribution mechanism for HD Map updates [15], which are more efficient in terms of data transmission and more effective in terms of update distribution than currently existing mechanisms. The focus of our research is thereby centered on the specific requirements and present side conditions of self-driving vehicles in contrast to human drivers. A self-driving car has to know its current route from the beginning to the end of the track or otherwise has to assume a most-probable path [35], which it will continue to follow. In contrast to a human driver this a priori knowledge can be used to develop *i)* a new distribution concept for the HD Map updates [15]. Furthermore the self-driving vehicle is expected to be equipped with several wireless transmission interfaces (e.g. WLAN and cellular). This allows it to *ii)* develop an intelligent selection mechanisms between the different transmission technologies [88] to further improve the distribution of HD Map updates in terms of overall network load and transmission costs.

Research Goal 2: Fast and Reliable Detection of Changes in the Road Infrastructure Based on Low-cost Widely Distributed Sensor Data

To develop new, intelligent sensor fusion techniques that enable the fast and precise detection of road infrastructure changes to ensure the reliability of the HD Map data and overall safety [89]. In our work we focused only on different kinds of low cost and largely deployed sensors. This includes cheap sensors, which are already available in most cars, but also in mobile devices such as smartphones and wearables, that might be carried by the passengers. Such sensors are not in the main focus of ongoing research for self-driving vehicles, as they do not directly enable the self-driving functionalities. However in our opinion the intelligent combination and the shear amount of such sensors is a currently not used, huge potential to support the general updating procedure of HD Maps. Not only in the initial phase, where only a minority of vehicles will have the full sensor equipment required for self-driving functions, but also in the long term, such sensors will provide meaningful insides into the relevant traffic information to improve the timely performance and robustness of map updates. The participation of older cars through mobile sensor devices can provide additional helpful information to the drivers themselves. That way the overall traffic safety as our combined research goal is further enhanced as well.

Research Goal 3: Robust, Optimized Provisioning of Data w.r.t Changing Environmental Conditions

To combine existing technologies and mechanisms and develop them further to improve the performance of network quality indicators provided to the self-driving vehicle to be used for their planning of future data transmissions [90]. This includes *i)* the identification of communication requirements for self-driving cars based on real-world data obtained during the Ko-HAF[3] project from a fleet of self-driving vehicles [91, 92]. As further contribution *ii)* the data-efficient creation of network quality maps (so called Connectivity Maps) by intelligently utilizing the transmission of non-time-critical data [93] is investigated. These obtained network quality indicators further are leveraged *iii)* for location specific training of machine learning algorithms for future throughput prediction [94, 95]. In all the developed techniques a special focus was set on the possible future deployability in terms of installation and maintenance costs. Furthermore *iv)* work on the combination of simulation and real measured data is conducted to facilitate the development of future complex vehicular communication simulation scenarios [96, 97].

Focus on maintenance loop of HD Map

In our work we specifically focus on the realisation of a robust and reliable data exchange to maintain the HD Map data for self-driving cars, as illustrated by Figure 1. For the data processing entity we are assuming in the following a central backend server entity, although we are well aware that the required functionality could also be realized through other entities e.g. distributed systems in the cloud [98] or the edge of the network [99]. A final decision therefore should be made especially with respect to the requirements of an actual large scale deployment such as reliability and scalability, but this is not the focus of this thesis. Furthermore in our work we focus on the collection of network quality indicators and their further post processing to provide meaningful background information for a subsequent data scheduling process. The scheduling process itself however, is out of the scope of our work, as it has been addressed comprehensively in previous work [90].

---

[3] www.ko-haf.de (Last accessed on August 1, 2019)

1.4 STRUCTURE OF THE THESIS

Following this brief introduction, we provide additional background on the different levels of automation specified for self-driving vehicles, as well as the available variants of communication technologies for them in Chapter 2. We discuss and classify related work relevant for the realisation of the update loop of the HD Map based on its identified three sub-components *i)* map update distribution, *ii)* map change detection and *iii)* reliable wireless provisioning in Chapter 3. Based on our discussion of related work on the map update distribution we introduce the *Dynamic Map Update Protocol* in Chapter 4. It enables the data-efficient distribution of map updates to self-driving cars based on their specific driving context as described in Section 4.1. The Dynamic Map Update Protocol is then further enhanced in Section 4.2 by the so called *HD Wmap extension*, that enables the self-driving vehicles to directly exchange data between each other via ad hoc communication.

In Chapter 5 we develop an algorithm to detect lane course changes based on low-cost sensor information. Its performance is evaluated on data obtained from a real highway scenario.

In Chapter 6 we present and evaluate, based on simulations and real word measurement data, new methodologies for the reliable provisioning of data in the context of the HD Maps. We begin with the identification of the actual communication requirements of self-driving cars in Section 6.1, based on the evaluation of the actual test fleet of self-driving vehicles of the Ko-HAF project on German highways. Outgoing from these requirements we develop and evaluate the ICCOMQS framework in Section 6.3, which obtains network quality information by leveraging the sent and received data required for the operation of the HD Map data loop. This geo-referenced connectivity information is then used to significantly improve existing approaches for future throughput prediction based on machine learning algorithms with a special focus on a low-effort large-scale future deployability. Additional contributions of our work focused on the development of fast and scalable simulation toolsets to support the future creation and evaluation of complex vehicular communication scenarios.

The thesis is concluded in Chapter 7 with a brief summary of the core contributions. Finally, we provide an outlook on potential future work.

2

BACKGROUND

In the following we provide an overview about relevant background information for our contributions in the domain of HD Map supported self-driving cars as motivated in Chapter 1. First we introduce the different levels of automation as keyed by the German Assocation of the Automative industry in Section 2.1. Secondly we provide an insight into the functionality of HD Maps and related Advanced Driver Assistance Systems (Sec. 2.2). Subsequent we provide insides into the domains of cellular network communication (Sec. 2.3), localisation (Sec. 2.4), mapping (Sec. 2.5) and simulation (Sec. 2.6) in correlation to our own scientific contributions of the upcoming Chapters.

2.1 SPECIFICATION OF AUTOMATION LEVELS

The German Association of the Automotive Industry (Verband der Automobilindustrie - VDA) has defined five different levels to classify the degree of automation for self-driving vehicles [2] (see Figure 70 in the Appendix). These levels range from Level 0: driver only, the driver has to perform all the common driving tasks himself, up to Level 5: driverless, the vehicle itself can handle all the driving tasks, a human driver is not required any more. Level 5 can be described as well with the term of a robotic taxi and therefore is considered as the final stage of automation, that can be reached. This degree of automation renders it possible for passengers, that might not be able to drive a car by themselves (elderly, disabled or blind people, ...) to participate in the vehicular traffic without relying on another person. True Level 5 autonomy however has not been reached until now (2019) from a technological point of view. Even Google's Waymo division, often considered as one of the most advanced researching groups in the field of self-driving cars, did not yet reach that point. Although recently launching their first self-driving taxi service in certain specific areas of Phoenix Arizona, they currently still rely on a safety driver during their rides [7]. Currently available in a commonly buyable production car are assistance systems of the Level 2: partly automatized. A level 2 car takes responsibility of steering and acceleration in certain driving environments. The driver however has to be fully aware of the surrounding traffic and must constantly supervise the system itself. Going further in terms of research all major participants, such as Google's Waymo division, ride-sharing-companies like Uber and the car manufacturers themselves, have reached or are currently aiming at the automation Levels 3: highly automated or 4: fully automated. Level 3 - highly automation means that the car can handle given driving tasks itself if a set of conditions is fulfilled. This includes the cars sensory equipment to work properly and the road conditions ahead on the vehicle's track to be well known. In the case of level 3 automation the human driver does not have to be constantly supervising the vehicle and the surrounding traffic. Instead the car has to inform him when it identifies that the safe driving conditions cannot be guaranteed any more. Then the driver is required to step in and take over the control of the wheel again. This so called handover has to be initiated by the vehicle itself. A safe handover procedure requires the vehicle to ensure safe driving for a certain period of time that the human driver has enough time to react and get in control again. In contrast to Level 3 (highly automation), where the driver is required to take over the wheel again, the self-driving vehicle of Level 4 (full automation) also has to ensure a safe driving state, should the driver not respond in time. This might include reducing the speed of the vehicle to reach a safe driving state. As a last resort a full stop of the vehicle, for example on the emergency lane of a highway, would be the consequence. In the following we will use the technical term of Level 3 - highly automation, when speaking of self-driving cars, as it is seen as the first level of automation that is truely providing benefits to the human [2, 100], as he does not have to constantly monitor the surrounding traffic environment.

Focus on Level 3 - highly automation

To ensure comfortable driving and a safe handover back to the human driver if necessary, the vehicles own on board-sensor equipment might not be enough for all possible driving conditions. There might be otherwise dangerous or uncomfortable driving situations, that can only be properly avoided with a foresight that goes beyond the vehicles own sensor range. A traffic jam, an accident behind a corner ahead or ice on the road are just some of such examples.

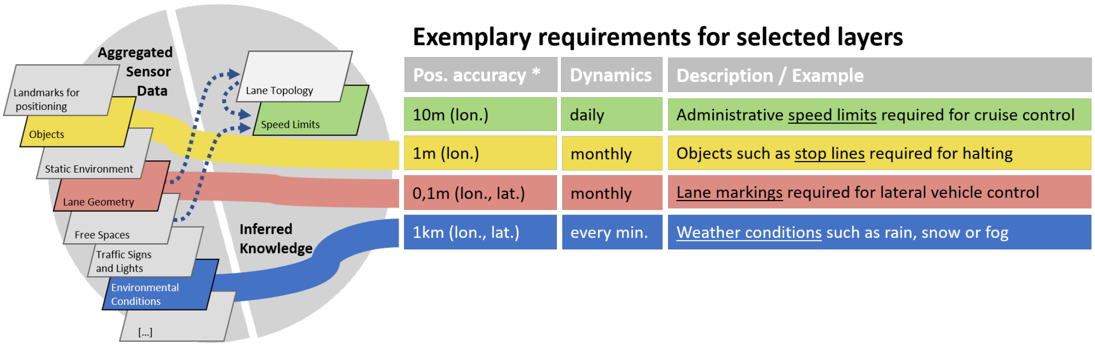
2.2 HD MAP AND RELATED ADVANCED DRIVER ASSISTANCE SYSTEMS

HD Map a precise virtual reference

To address these issues the highly automated vehicle thus further relies on an additional ″virtual″ sensor for its driving task, the so called High Definition Map (HD Map) [9–11]. This map provides highly accurate geo-referenced information (up to the centimetre-level of precision) to the car in comparison to nowadays standard navigational map data (with a localization accuracy of several meters). The HD Map thus can be seen much more as a highly precise virtual 3D representation of the actual real world [13].

> ″Really, [our maps] are any geographic information that we can tell the car in advance to make its job easier. ... We tell it how high the traffic signals are off the ground, the exact position of the curbs, so the car knows where not to drive. We'd also include information that you can't even see like implied speed limits.″ (Andrew Chatham, the Google self-driving car team's mapping lead 2014 [13])

As introduced in Section 1 the HD Map as centralized data source shared between all self-driving vehicles, e.g. via a central Backend Server, maintains its own functionality by receiving a continuous stream of traffic information from different data sources. This includes federal organisations, such as the police or road authorities, but also the sensors of the self-driving vehicles themselves. By sharing their personal sensor readings with the central HD Map the self-driving vehicles enable the generation of



Exemplary requirements for selected layers

Pos. accuracy *	Dynamics	Description / Example
10m (lon.)	daily	Administrative speed limits required for cruise control
1m (lon.)	monthly	Objects such as stop lines required for halting
0,1m (lon., lat.)	monthly	Lane markings required for lateral vehicle control
1km (lon., lat.)	every min.	Weather conditions such as rain, snow or fog

Figure 3: Different layers of the HD Map, with varying time and accuracy requirements. ©Ko-HAF

map updates on a much faster timely level, compared to the other available sources. That way map updates can be provided with a high frequency to the vehicles internal copy of the map. Furthermore the vehicles can incorporate information from other vehicles into their driving task, which goes beyond the range of their personal sensors. For example the first car detecting a new obstacle on the road ahead can quickly warn all following vehicles, long before they reach the designated area, which as stated enhances the traveling comfort and overall safety. As illustrated in Figure 4 the raw sensor data obtained from the different components inside the vehicle (GPS, camera, radar, lidar, ...) have to be pre processed, before they can be send to the Backend Server (in the following called Safety Server, due to the safety related functionality of the HD Map). The geo-referencing of the newly detected features in the vehicle's onboard HD Map allows the Safety Server to effectively detect changes in its own map data. To efficiently access the various geo-referenced objects inside the map to update and maintain them, the HD Map is structured in different layers (see Figures 3 and 4). These different layers are organized in terms of the timely dynamics and the required accuracy of the contained objects. Traffic events such as the change of a variable traffic sign's value or the release of the side strip for a temporary optimisation of the dense traffic might change their own status just within minutes of time. Lane markings require a precise localisation in the map down to the level of centimetre accuracy for the self-driving vehicles to be able to localize themselves and steer correctly. Weather conditions such as fog, rain or snow in contrast span over several kilometres of the track and thus do not require such a high localisation precision. Further details about the HD Map concept and an in depth description of an actual prototypical realization are provided in Chapter A.2 of the Appendix.

Continous data exchange to maintain HD Map functionality

In the domain of map based Advanced Driver Assistance Systems (ADAS) several other technical terms, besides the HD Map, have been coined. This includes the electronic Horizon (eHorizon) [35, 101], the ADAS Horizon [34, 102] and the Local Dynamic Map [30–33]. All of those terms poses a fluent transition between each other and are often used synonymously. This makes it especially difficult to differentiate sharply

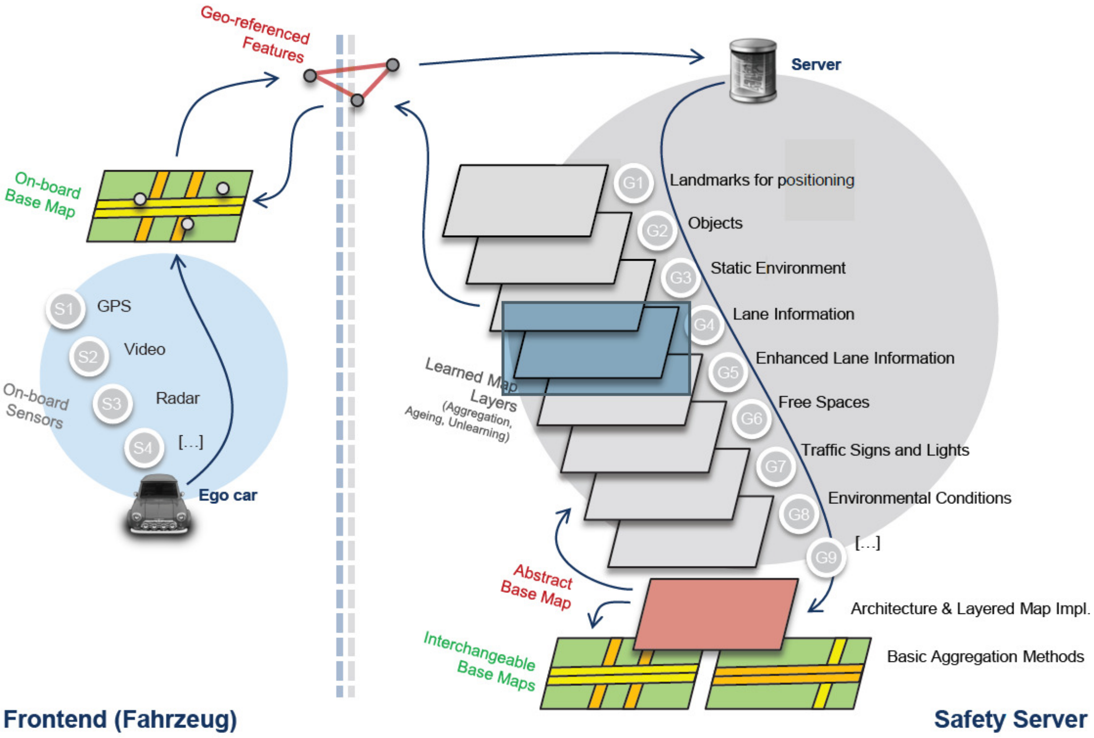

Geo-referenced Features
Server
On-board Base Map
S1 GPS
S2 Video
S3 Radar
S4 [...]
On-board Sensors
Ego car
Learned Map Layers (Aggregation, Ageing, Unlearning)
G1 Landmarks for positioning
G2 Objects
G3 Static Environment
G4 Lane Information
G5 Enhanced Lane Information
G6 Free Spaces
G7 Traffic Signs and Lights
G8 Environmental Conditions
G9 [...]
Abstract Base Map
Architecture & Layered Map Impl.
Interchangeable Base Maps
Basic Aggregation Methods
Frontend (Fahrzeug)
Safety Server

Figure 4: Abstract working principle of the HD Map. The readings of the various sensors inside the self-driving car are correlated with the vehicles present map data. That way geo-refrenced features (e.g. detected lane markings and traffic signs) are derived and wirelessly transfered to the map hosting server. The server itself takes several of these features to generate updates for the map data, which then is provided back to the self-driving vehicles. ©Ko-HAF[4]

between their conceptual emphases. To give the reader a better overview about such map based Advanced Driver Assistance Systems and to put them into the context of our own work, a short clarification of the different terms is provided in the following. The key connecting component in all those systems is some kind of map data source, which is used to facilitate the general driving task. This could be a normal navigation map designed for human drivers, our as in our case HD Maps, specifically created for self-driving vehicles and their driving task. In other words the HD Map can be seen as fundamental basis, whose data is then further conditioned or enhanced to support the aspect of safety during the ride. The electronic Horizon (eHorizon) and the ADAS Horizon are two different terms for the same general technical concept, which are used in parallel to each other. The Horizon emphasise the extraction and preparation process of relevant traffic data out of a digital map specifically for the vehicles current track. This map data is further combined and enhanced through the cars own on board sensor readings, e.g. its current location and driving speed. The HD Map can be used as such a map data source. If the electronic Horizon is used to provide specific traffic

---

[4] https://ko-haf.de/fileadmin/user_upload/media/abschlusspraesentation/14_Ko-HAF_Continuous-Updating-of-Backend-HD-Map-Data.pdf (Last accessed on August 1, 2019)

information to a human driver, as supposed during its initial development phase, it cannot assume a given route. Instead it is utilising a so called Most Probable Path calculation [101, 102], to anticipate the most likely driving behaviour of the human driver. This is in contrast to the self-driving vehicles (see our own contribution in Section 4), which always requires a given specific navigation route for them to perform their driving task. The Local Dynamic map is an Advanced Driver Assistance System standardized by the European Telecommunications Standards Institute (ETSI) [30]. In contrast to the Horizon approach, that enhances the navigation data of the car through additional sensor readings from the vehicle itself, the Local Dynamic map further achieves an enhancement of the data through direct ad hoc communication between the vehicles in close proximity. That way the cars are able to exchange relevant information about the current traffic condition, including their own position, traveling speed and driving direction. Similar to the different layers defined for the HD Map, the ETSI has categorized real world traffic objects with different degrees of timely relevance into four separating layers of data for the context of the Local Dynamic Map. These four different layers are: *i)* permanent static data, map data, provided from a map supplier, *ii)* transient static data, e.g. speed limits, *iii)* transient dynamic data, e.g. weather situations and traffic information, *iv)* highly dynamic data, e.g. the surrounding vehicles speed and driving direction. However the ETSI states that the Local Dynamic Map itself does not contain any kind of information of the layer *i)*. This is due to the reason that navigational map data is not required by all Intelligent Transportation Systems (ITS) applications. If required by an application, e.g. our considered self-driving vehicles, it is assumed to be provided by an interchangable map data provider, which is not part of the standard specification. The data contained in layer *iv)* is achieved through continuous information exchange between the vehicles, regarding their current driving behaviour, via ad hoc communication using so called Cooperative Awareness Messages (CAMs), that provide the vehicles current position, driving speed and direction. This highly dynamic layer cannot be stored in the HD Map and be provided through a back end entity (e.g. our Safety Server), as its timely relevance is too short and the information is only relevant for the specific vehicle. Consequently it is stored locally in the car, thus the name of Local Dynamic Map.

In conclusion the ETSI definition of the Local Dynamic Map does not include the first layer of the HD Map data, but rather contains a further highly dynamic layer, which is achieved via direct ad hoc communication between the vehicles.

2.3 IMPORTANCE OF THE CELLULAR NETWORK FOR SELF-DRIVING VEHICLES

Self-driving vehicle are considered important use case of 5G

In the perspective of the ongoing development of future 5G cellular communication technologies [52, 53], the self-driving vehicle represents an important use case to be considered. The 5G PPP consortium identifies three main 5G service types in the Metis II project [52], all with different requirements regarding the achievable network performance. As illustrated by Figure 5 they are: extreme mobile broadband (xMBB), massive machine-type communications (mMTC) and ultra-reliable machine-type communications (uMTC).

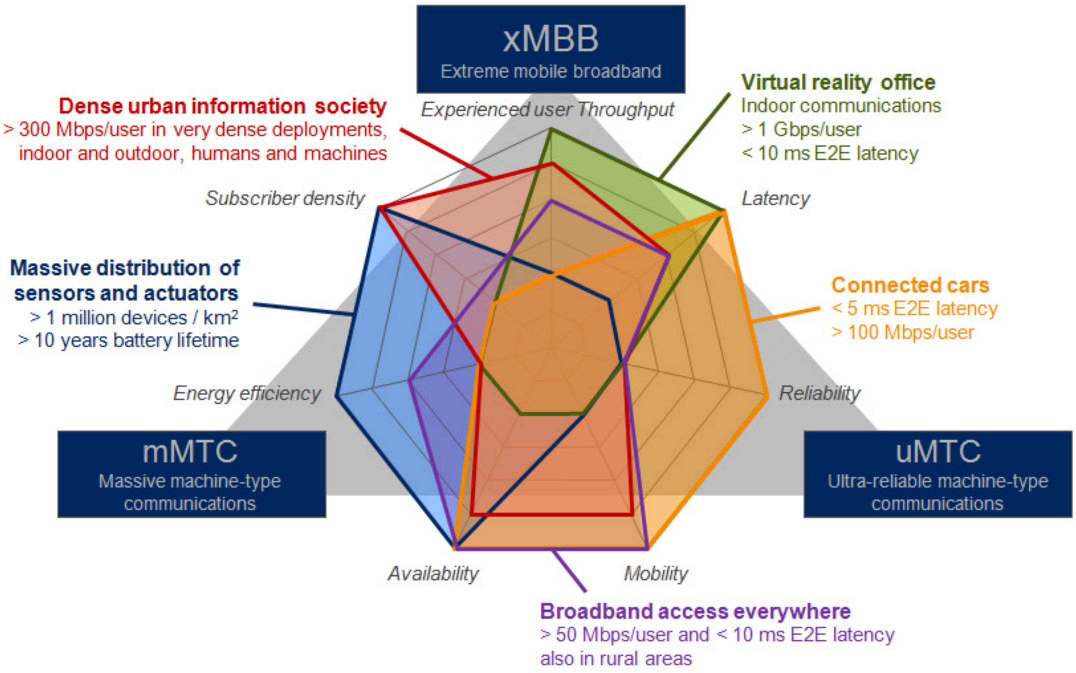


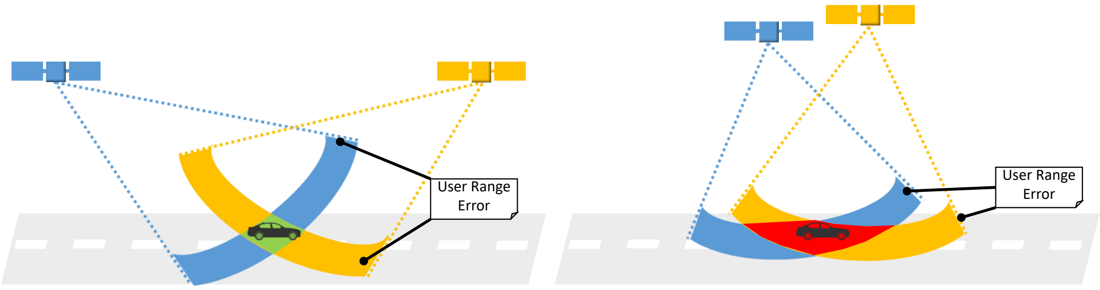
Figure 5: Main 5G service types and representative use cases considered by the 5G PPP consortium in the METIS II project. (Source: Fig. 1 in [52])

The requirements of self-driving vehicles supported by an operational HD Map lie in the area of ultra-reliable machine-type communications (uMTC). Having similar requirements as identified by the two use-case domains of "Broadband access everywhere" and "Connected cars", as self-driving cars might also roam in rural areas, with a comparable sparely deployed cellular network infrastructure, requiring a continuous data exchange to keep their HD Map operational. These two domains have a strong emphasis in the requirement areas of network availability, mobility of the end devices (the cars themselves) and reliable data transmission.

To map those general requirements on measurable quantities, several different key performance indicators have been specified in the general domain of network communication as well as specifically for the cellular network communication itself. Section A.3 provides details about the set of indicators relevant for our work as described in Chapter 6.

2.4 POSITION DILUTION OF PRECISION OF GLOBAL NAVIGATION SATELLITE SYSTEMS

To know its exact location on the road is a highly important information for a self-driving car. One of the most important sensor sources to solve this task is the measurement data received from Global Navigation Satellite Systems (GNSS). The general working principle of GNSS is explained in the following, as far as it is relevant for our personal work presented in Chapter 5.



(a) Good position of the satellites relative to the position of the vehicle - low Position Dilution of Precision

(b) Bad position of the satellites relative to the position of the vehicle - high Position Dilution of Precision

Figure 6: Impact of the Dilution of Precision on the location estimate due to the constellation of the visible satellites (green - low / red - high). (Inspired by [103] and [5])

At least four satellites are required for localization. ...

... more improve overall position estimation accuracy.

Dilution of Precision due to position of satellites.

Each GNSS satellite circles around the earth in an exactly known orbit. Therewhile it continously broadcasts a reference timing signal by relying on several inbuilt atom clocks. The receiving end device's sensor equipment then estimates the travel time between itself and all sending satellites it can obtain a signal from to derive its own position. This measurement can only be executed with a certain margin of error, the so called User Range Error (see Figure 6). Through an increase in the number of received reference signals (more visible satellites), this error can be reduced by averaging out the individual errors to each other. Besides the overall number of visible satellites also their relative position to each other is important for the achievable position estimation accuracy. This so called Dilution of Precision (DOP) is illustrated by Figure 6. The area, which has to be considered as the possible location of the receiver due to the User Range Error (e.g. the black vehicle in the image), is varying with the location of the satellites. Satellites that are distributed in a favourable way (Figure 6a) achieve a low Position Dilution of Precision (indicated through the green, rectangular area). Satellites, which are very closely located to each other experience a high Position Dilution of Precision and thus only achieve a poor location estimation (indicated through the red, rectangular area in Figure 6b).

2.5 GEOGRAPHIC LOCATION VIA GEOHASHES

The geographic localisation index referred to as Geohashes was first presented by Niemeyer[6] [104] in 2008. It is a public domain geocoding system that enables the efficient mapping of latitude and longitude coordinates (WGS 84) via a hash-function into a string identifier. Depending on the size of the string (it's amount of letters) the Geohash represents a bounding box with a respective size that contains the specified coordinates (Fig. 7). The size of the bounding box thereby ranges form several thousand

---

[5] https://www.gps.gov/systems/gps/performance/accuracy/ (Last accessed on August 1, 2019)

[6] https://web.archive.org/web/20080305223755/http://blog.labix.org/ (Last accessed on August 1, 2019)

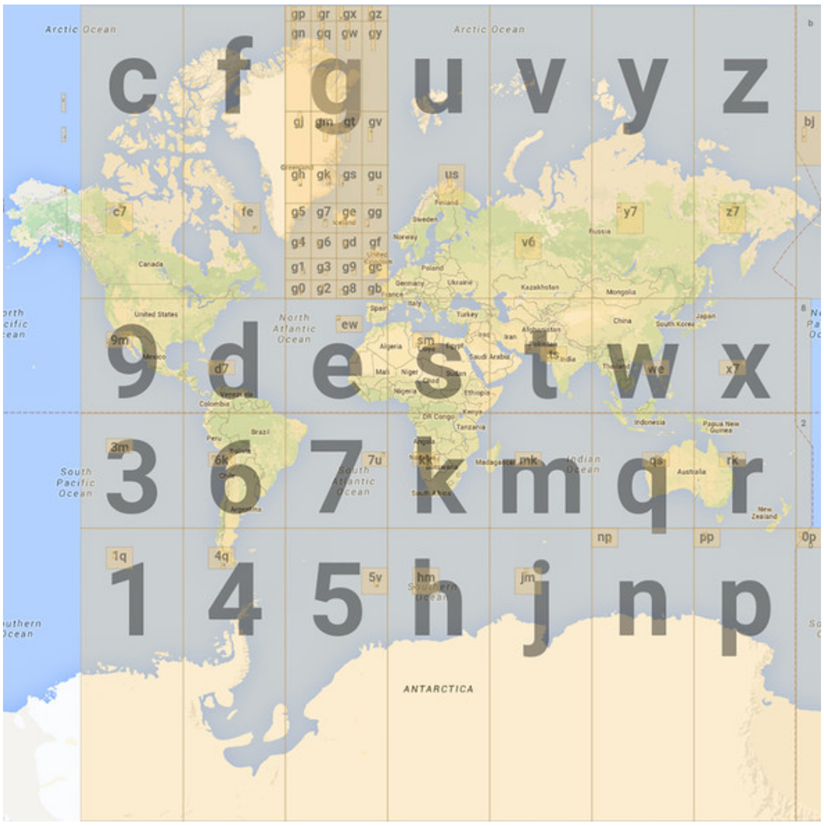


Figure 7: Example visualization of nearby Geohashes of different accuracy levels. The higher accuracy is indicated by longer strings. Neighbouring strings start with the same sequence of letters. (Source: https://www.movable-type.co.uk/scripts/geohash.jpg (Last accessed on August 1, 2019))

square kilometres down to only a few centimetres in size, which realizes a localisation approach that satisfies various accuracy requirements. Geohashes are a furthermore supportive indexing strucuture, as closely located areas share the same initial letters of their defining string. Consequently Geohashes are used in various digital applications related to georeferenced information, for example mapping and communication [105].

2.6 SIMULATION OF URBAN MOBILITY - TRAFFIC SIMULATOR - SUMO

SUMO (Simulation of Urban Mobility) [106] is a vehicular traffic simulation software developed at the Institute of Transportation Systems at the German Aerospace Center (DLR)[7]. SUMO enables the physically correct simulation of individual vehicles (acceleration, braking, obeying speed limits and traffic rules, ...) of different classes (e.g. sedans, trucks, bikes ..., as shown e.g. in Figure 8) at a large scale (several thousand vehicles at once).

---

[7] https://sumo.dlr.de/index.html (Last accessed on August 1, 2019)

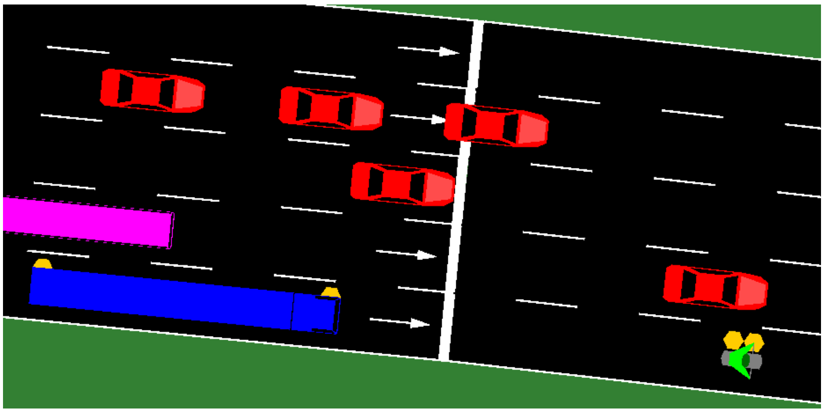


Figure 8: Visualisation of SUMO's[8]capabilities to simulate various different vehicle types (e.g. sedans, buses, trucks and bikes) on a per vehicle basis with correct movement patterns (e.g. physically correct acceleration and braking as well as obeying the traffic rules).

SUMO is applied in many related scientific contributions (e.g.[90, 107, 108]) as the availability of various traffic scenarios provided by the scientific community (e.g. [109],[110]), which resemble real world scenarios [9] makes it a reliable and well accepted software resource. As result we also rely upon SUMO for our simulation scenarios of vehicular traffic for the scenarios (e.g. Sections 4, 6.3, 6.6.1 and 6.6.2, which we cannot replicate through our actual real world tests due to scalability reasons (see Section A.11 for further details regarding the simulated scenarios).

After this initial overview about the scientific terms relevant for the context of our work we now present related scientific publications in Chapter 3.

---

[8] https://sumo.dlr.de/index.html (Last accessed on August 1, 2019)

[9] https://sumo.dlr.de/wiki/Data/Scenarios (Last accessed on August 1, 2019)

3

RELATED WORK

Accordingly to the working cycle of the HD Map as described in Section 1.1 and its three main building blocks Generation, Distribution and Provision as described in Section 1.1, we present related scientific contributions in the following.

Generation, Distribution, Provision

Namely they are: i.) the distribution of digital navigation map updates (Sec. 3.1), ii.) the aggregation of vehicular sensor data to generate map updates (Sec. 3.2) and iii.) concepts to optimize the provision of both of these data streams via wireless communication (Sec. 3.3). In Section 3.4 we provide a summarizing discussion of the Related Work and outline the identified research gaps that are addressed in this thesis.

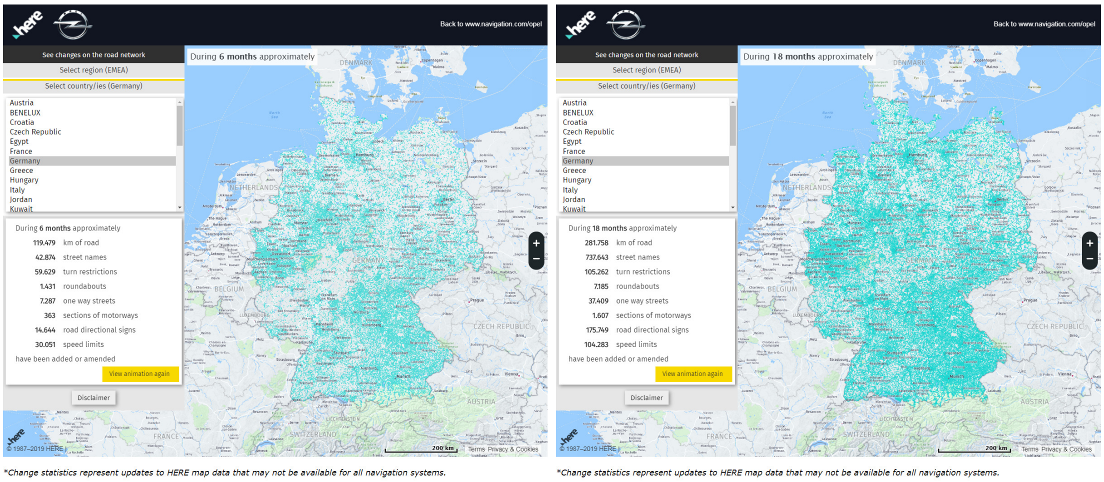
HD Maps are highly relevant to ensure safe and comfortable driving capabilities of self-driving vehicles. Due to the density of the stored geo-referenced information and its location accuracy (sub-meter level) the HD Map database needs to be updated much more frequently compared to the update cycle of common navigation map material to maintain its functionality. An HD Map for example requires updates in the time frame of minutes to hours, e.g. to incorporate variable traffic signs and moving construction works, whereas a normal navigation map receives updates only after some months, for example to include more static road changes such as new permanent speed limits. From the following overview about related scientific publications for map updates we derive the necessity for new maintenance techniques to ensure the successful operation of HD Maps.

3.1 MAP UPDATES FOR DIGITAL NAVIGATION SYSTEMS

Online and offline navigation systems

There exist two major variants of navigation systems: offline and online. Offline navigation systems on the one side are most commonly directly integrated into the vehicle. These systems operate offline by storing their complete navigation map in a single binary data file represented via an often propretary physical storage format (PSF). The major advantage of such a propretary binary storage is its efficent accessebility through specialized algorithms [24] to retrieve the desired route.

The binary data representation however, also is the systems major disadvantage. The hierarchical structure of the data and the multilevel connectivity between nodes and links, representing the street infrastructure and related points of interest (POI), renders the map data incapable of gradual updates [24]. Consequently the binary storage file always has to be completely replaced when introducing new updates into the road infrastructure of the map database (e.g. new road kilometer, speed limits, road directional signs, ...). In consequence several gigabytes of data [25] have to be exchanged. The map replacement procedure is often a manual and time-consuming process (e.g. several minutes up to an hour [25]), which in the meanwhile completely



During **6 months** approximately

During **6 months** approximately
119.479 km of road
42.874 street names
59.629 turn restrictions
1.431 roundabouts
7.287 one way streets
363 sections of motorways
14.644 road directional signs
30.051 speed limits
have been added or amended

**Change statistics represent updates to HERE map data that may not be available for all navigation systems.*

During **18 months** approximately

During **18 months** approximately
281.758 km of road
737.643 street names
105.262 turn restrictions
7.185 roundabouts
37.409 one way streets
1.607 sections of motorways
175.749 road directional signs
104.283 speed limits
have been added or amended

**Change statistics represent updates to HERE map data that may not be available for all navigation systems.*

(a) after 6 months

(b) after 18 months

Figure 9: Amount of changes (e.g. new road kilometer, speed limits, road directional signs, street names, turn restrictions, ...) introduced into Here's digital navigation map for the area of Germany over a time of 6 and 18 months (map data as of 14.01.2019). ©Here and Opel Automobile GmbH[11]

disables the navigation system's functionality. To reduce the amount of these complex updating processes, mapping companies (e.g. TomTom[10] and Here [25]) nowadays are commonly gathering all new map changes over a period of several months before they provide a map replacement to the customers. This delay of updates leads to a degradation of the maps routing capabilities in the meanwhile, as the road network is subject to continuous changes (e.g. construction work, new traffic regulations, ...). Figure 9 illustrates the changes (green markings) introduced into map supplier Here's database of Germany over a time period of 6 and 18 months (see Section A.12 for further details). During a period of six months (map data as of 14th January 2019) 119.479 kilometer of street including 363 sections of motorways and many further changes were added or amended in the map database. After 18 months these numbers increased to 281.758 kilometer and 1.607 motorway sections respectively. Such numbers are similar for other regions world wide, for example as described in the work of Ashara et al. [29] for a common navigation map of Japan. Here itself claims to introduce about 2.7 million vehicular traffic related changes into their global map database each day [111].

Road network continuously changes

Online navigation systems on the other side are mostly represented through applications installed on mobile devices, such as smartphones. These systems don't permanently store their map data, but instead always obtain it via a wireless connection from a central map maintaining entity (e.g. a central server). Future requests of the same route trigger the process anew, which results in redundant data transmission and additional (e.g. cellular) transmission costs. Furthermore, such systems are

---

[10] https://www.tomtom.com/en_gb/sat-nav/maps-services/map-updates/ (Last accessed on August 1, 2019)

[11] http://mapchanges.navigation.com/?app=opel (Last accessed on August 1, 2019)

incapable of performing the routing task if no data connection is available, a possible situation in our vehicular scenario.

In summary both concepts contain certain disadvantages for the application in our considered vehicular scenario. To overcome these problems related scientific work has been conducted as explained in the following Section 3.1.1.

3.1.1 *Partial and Incremental Map Updates*

To overcome the previously mentioned disadvantages of offline and online navigation systems new techniques have been developed in various scientific contributions [22–24, 28, 29, 81–83] to improve the previously described map updating process. The two most prominent techniques in that domain are the partial and incremental map updates.

Map tiles

To realize partial map updates (as described for example in [29, 81, 82]) the whole area, which is covered by the map data, is split up into smaller individual tiles (see Fig. 12 for illustration). Each tile then is treated as a distinct separate map and in consequence is updated individually from each other. In doing so the updating algorithms have to ensure the routing consistency in between the different map tiles by updating further surrounding map tiles if necessary. Specific algorithm to solve this problem for example were developed by Asahara et al. [29] as well as proposed by researchers of Hitachi [28]. Incremental, consecutive map updates [22, 23] in contrast to a full map replacement enable the application of a sequence of updates to the currently present map on the client's side. Therefore a central map maintaining data server (or similar data processing entity) keeps track of the occurring road infrastructure changes, which affect the map. Based on the current version of the map stored on the client's internal memory the central server then is able to generate a specific update file for the client. This file has a comparable small data footprint, as only map changes have to be communicated to the requesting client. Most commonly both techniques, partial and incremental map updates, are used in combination. The individual map tile updates are then provided from the central server via wireless communication to the requesting vehicles and stored in their internal database for further future usage. That way the advantages of offline and online navigation (continuous available navigation functionality and always up-to-date map data) are combined together in one system. Most of the presented approaches focused on the context of common navigation maps as used in nowadays manual-driven, production vehicles. Only Bastiaensen et al. [81] mentioned self-driving cars as a possible usage scenario, but did not investigate specific update concepts for such machine-driven vehicles.

Sequential updates

Besides an infrastructure based communication, vehicles can also directly communicate with each other to exchange data of any kind. Enabling ad hoc communication technologies and related scientific Work are described in the following Section 3.1.2.

3.1.2 *Ad Hoc Vehicular Communication for Data Exchange*

To realize the physical communication channel between two adjacent vehicles there currently exist several different technologies.

Most commonly the data exchange is either realized via WLAN (e.g. IEEE 802.11p [112–115]) or cellular communication technology (summarized as C-V2X e.g. LTE-V2X [116] and 5G-V2X [117, 118]).

Ad hoc communication via WLAN or cellular technologies

A third concept currently investigated and discussed in the scientific community is the usage of visible light communication [119–121] to exchange data in the vehicular communication context. Therefore the LED front and rear lights of the vehicles are used to exchange high frequent light pulses, which are then received by photo detectors as receiving counter parts. All these technologies are for example used to realize further advanced safety applications [58]. Therefore each vehicle exchanges information with its surrounding neighbors in close proximity via broadcast messages. Common examples for messages to be exchanged in this context are the Cooperative Awareness Message (CAM) [59] and the Decentralized Environmental Notification Message (DENM) [60]. The DENM message is used for example to warn other vehicles about road hazards, dangerous weather and obstacles on the road, e.g. debris. Via the CAM message the vehicle shares for example its current position, type, speed and heading direction. Besides driver warnings these messages can be used in the context of maps to realize the so called "Local Dynamic Map" (LDM) [30], which was standardized by the European Telecommunications Standards Institute (ETSI).

Local Dynamic Map

The Local Dynamic Map provides vehicles with detailed information regarding their current surrounding traffic environment, such as the position and speed of all neighbouring cars (via CAMs) or further ahead road obstacles (via DENMs). Depending on their dynamic and timely relevance the various types of data are grouped into different map layers (see Figure 1 in [31] for illustration and [30] for further details) However the ETSI did not further specify the properties of the base navigation map to be used in the Local Dynamic Map, because all objects and events are locally referenced by WGS 84 coordinates (latitude, longitude). In their proposal any digital map is applicable to host the described dynamic layers on top. This circumstance lead to several further research proposals, which addressed certain issues in the ETSI specification [32] or proposed a concrete implementation of the LDM [31, 33] (e.g. using OpenStreetMap map data as base layer) often in combination with a performance evaluation of the system itself. Besides the mentioned safety applications also a multitude of scientific contributions exist, which address the more general task to offload otherwise costly data transmission from the cellular network onto the free of charge WLAN communication channel [54–57, 122–126].

Many of the proposed concepts thereby aimed at the task to share highly individual data streams, such as video or audio streams, via ad hoc communication between the vehicles. As a result several of the authors [54, 55, 123] claimed the necessity of further infrastructure to enable the communication. Therefore additional so called Road Side Units (RSUs) should be installed. These RSUs operate as data beacons to relay the data streams between two cars, which would otherwise be out of communication range.

Nagel et al. [79] propose a concept to predict the movement patterns of the vehicles and develop a new wireless channel model to identify the possible best points for a data exchange. Their approach is closely related to the concepts discussed in Section 3.3, which aim to achieve similar improvements for the context of infrastructure-managed wireless communication.

With the distribution of new map updates also arises the question, how such updates can be efficiently generated to ensure the full operational cycle of the HD Map (Figure 1). This general problem and the related solution approaches are presented in the following Section 3.2.

3.2 VEHICULAR SENSOR DATA AGGREGATION TO MAINTAIN MAP DATA

Initially HD Maps for self-driving vehicles are generated by dedicated mapping vehicles [127, 128] (see Figure 10). Such cars can rely on a set of top of the line sensors including lidar, radar, differential GPS and camera systems to build the HD Map. In consequence they are costly to operate and only a few in numbers.

Special mapping vehicles initially

Due to the high precision (sub-meter level) and density of geo-located information contained in the HD Map [11, 18, 129–131] the continuous changes of the road network infrastructure (e.g. through construction works, accidents, new traffic regulations, ...) [14] render the ongoing maintenance of the HD Map a much more challenging task compared to common digital navigation maps used by human drivers. Self-driving vehicles constantly rely on the HD Map to retrieve up-to-date traffic information, which is often only valuable during a rather short period of time (e.g. the current status of variable traffic signs, weather conditions such as black ice on the road, the location of moving roadworks and the according change of the road curvature, ...) to perform their driving tasks with high safety and comfort. Consequently, the small amount of special mapping vehicles cannot solve the maintenance task alone, as the appearance frequency of new traffic events is simply too high to be fully covered by them. A large variety of concepts have been proposed and different sets of sensors have been developed [132] to address the general problem of map maintenance. In our research context of self-driving vehicles one of the most promising ideas addressed in various scientific concepts to improve the required time to produce a map update is to rely on sensor equipment, which is already available in cars. These different sensor sources, which are moving around with the vehicles are summarized through the technical term Floating Car Data (FCD) in the following.

Floating Car Data (FCD)

Depending on the desired accuracy of the map updates (e.g. location accuracy on road or lane level) the different approaches rely on a wide selection of sensors. The used data sources range from low-cost sensors available in common production vehicles [44, 133] or mobile devices, such as smartphones [46, 134, 135] up to the high-end sensor equipment used in self-driving prototype vehicles [128, 136], which is comparable to the setup used by the dedicated mapping vehicles.

Wide sensor variety

In the following, we provide an overview about the investigated Related Works arranged accordingly to their achieved accuracy and used sensor equipment. The first group of approaches achieves the generation of road accurate maps.

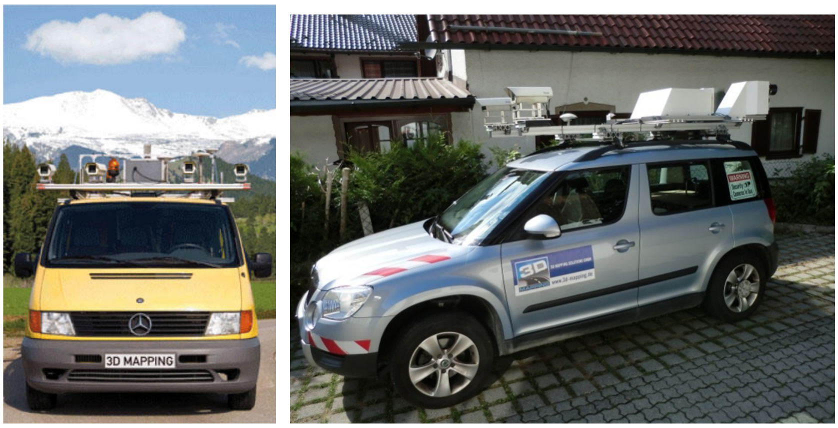


Figure 10: Mapping vehicles used for the initial creation of the HD Map used during the Ko-HAF Project. ©3D Mapping Solutions GmbH.

3.2.1 *Road-level Map Generation with Floating Car Data*

Brüntrup et al. [137] and Niehöfer et al. [134] independently from each other propose both a client/server-based architecture to derive the road infrastructure of an unknown area from a set of collected GNSS traces. The GNSS traces can either be provided by vehicles themselves or through mobile phones transported in the cars. The traces are initially preprocessed to remove outliers based on the achieved average speed and acceleration. The proposed concepts are either used to (i) incrementally update and refine an existing road or to (ii) create a completely new road in the map.

Cao et al. [138] present an approach, which realizes incremental updates of already identified road center lines by calculating so called energy wells for the newly incoming GNSS trace data. The authors thereby rely on a large fleet of taxi cars to obtain their evaluation data set.

Davies et al. [139] present a similar framework to build road-level accurate maps from Floating Car Data. Via a mathematical analysis the authors state that under the assumption of an average standard deviation σ of a common low-cost GNSS device between 3.5 and 4.5 meter (based on the publications of McDonald [140] and Prasad [141] and the central limit theorem) that at least 73 GNSS traces are necessary to differentiate between two adjacent roads (not lanes) to compensate for the Gaussian Noise inherent in the GNSS traces.

A comprehensive overview of the aforementioned approaches and further works is presented by Ahmed et al. [142]. The authors provide Open Street Map benchmark material and evaluate the runtime performance of the different proposed algorithms, achieving times between several minutes and several hours.

The following contributions create lane-level accurate maps. From our working experience in the German research project Ko-HAF (Cooperative Highly Automated Driving) (see Section A.2) this accuracy level is considered as the bare minimum required for the HD Map data to ensure a safe localization and thus operation of the self-driving vehicle within the ongoing traffic.

3.2.2 *Lane Accurate Map Generation*

Betaille et al. [143] generate lane accurate maps by fusing precise high accurate kinematic GPS (PPK) sensors with dead reckoning movement estimations of the vehicle. By relying on the expensive kinematic GPS the authors achieve highly accurate results, resembling the performance of the dedicated mapping vehicles.

Chen et al. [144] use a Gaussian Mixture Model to achieve lane-accurate results in the clustering of collected GNSS trace data. Their initial concept to structure and preprocess the data is illustrated by Figure 23a and described in detail in Section 5.1.1. It is used similarly in several other works [46, 135, 145]. Chen et al. focused their work on the lane accurate mapping of intersections. They evaluated their concept by relying on GNSS data provided through a large fleet of 55 vehicles, which were equipped with common GPS loggers (comparable to the ones available in nowadays mobile devices). They did not investigate changes of the road network, but stated the necessity therefore (e.g. for the detection of construction sides) as future work.

Uduwaragoda et al. [135] and Neuhold et al. [46] perform the same preprocessing step of the GNSS traces, which they collect from smartphones transported inside cars. In the following clustering procedure in contrast to [144] both authors chose the Kernel Density Estimation algorithm to identify the center lines of each lane. To achieve lane-accurate results the two approaches have to rely on a large set between 80 and 200 traces, depending on the investigated scenario and the used GNSS sensors. Neuhold et al. [46] further assume side conditions based on federal regulations regarding the necessary minimum width between two lane centers. These side conditions however might not be fulfilled in all conceivable scenarios. A construction side for example can necessitate the reduction of the available lane width.

Joshi et al. [128], Guo et al. [44] and Massow et al. [133] are three further examples of approaches, which build lane accurate maps by relying on additional sensor data to be fused with collected GNSS traces. This includes sensors like the camera [44, 133], lidar [128] or radar [133], as well as information provided by orthographic images [44] and coarse maps [128]. Massow et al. [133] further state the varying quality in the achieved GNSS traces as a problem to be considered, but do not address it any further in their work. Similar to the work of Betaille et al. [143], these publications [44, 128, 133] motivate the use of additional sensors data to be fused with GNSS traces to improve the overall quality of the sensor data (reduce noise, remove outliers, ...). Unfortunately due to the additional and sometimes very high costs of these sensors they are currently not commonly included in all production vehicles. Especially older vehicles, which still drive on the streets cannot provide such sensor readings. This leaves out a large group of potential providers of safety relevant sensor information. Furthermore several external influence factors can have a significant negative impact on the achieved overall sensor performance. Chen et al. [48] stated several of them for their camera based approach. This includes weather effects (e.g. sun blinding, rain, snow) and the day and night cycles. Similar influencing factors can be identified for lidar and radar sensors.

The presented works in the following all have in common to rely on low cost sensors to derive further knowledge about the vehicles current position and status on the road. They are ubiquitously available in vehicles as well as mobile devices. Consequentially, we see a huge potential in their further usage to maintain maps via Floating Car Data.

3.2.3 *Lane Identification with Low Cost Sensors*

Derive car position from steering maneuver

Aly et al. [47] and Wu et al. [49] propose comparable approaches to identify the current lane in which a traveling vehicle is located. By using the readings of inexpensive accelerometer and gyroscope sensors, which are available in nearly all production vehicles and mobile devices, the authors can identify the orientation of the vehicle and detect the execution of lane changes. Through a series of such lane changes (see for example Figure 24), the authors are able to identify the vehicles current lane. Knowledge about the initial lane of the vehicle is not required. Instead a Markov localization model or a Gaussian probability distribution is used to keep track of all possible initial lane positions. Possible initial positions then get narrowed down with an increasing amount of executed lane changes. In the end only a single possible current lane remains. Aly et al. further suggest to rely on so called boostrap anchors (conditions on the road - e.g. a pothole) and organic anchors (traffic rules) to improve the performance of the detection algorithm. Both algorithms achieve a robust lane identification performance between 80% and 86%.

As stated previously Chen et al. [48] identified several negative influencing factors on camera-based sensor systems. In their work, the authors compared the performance of a camera-based detection algorithm for steering maneuvers with an approach that similarly to Aly et al. [47] and Wu et al. [49] relied on the accelerometer, the gyroscope and the magnetometer as main source of sensor input. Through the comparison Chen et al. were able to demonstrate the robustness of this approach, as it was not effected by the stated weather effects, which hindered the camera-based algorithm.

Ahmed et al. [51] suggest that through the combination of the mentioned low-cost sensors and the cars own on-board sensors via an On-board Diagnostic Interface (OBD) driving maneuvers and the vehicles trip direction can be identified. The authors evaluate their concept in an intersection scenario and achieve a detection performance of 93% for the maneuver identification and 89% for the trip identification. The work of Liu et al. [146] is another publication that supports the smartphone as reliable sensor platform. Solely based on the sensor information provided by the smartphones of bus-passengers the authors predict arrival times and draw further conclusions on the currently ongoing traffic.

To retrieve map updates and send vehicular sensor data the car can rely on different wireless data links. Most of them can be grouped into two different categories. On the one side, the vehicles can directly communicate with each other via ad hoc communication, as introduced in the previous Section 3.1.2. On the other side, the self-driving vehicles can rely on infrastructure-managed communication technologies. The cellular network is probably the most prominent and well established example of this kind. The Ko-HAF project for example relied on cellular network communication for the maintenance of the HD Map data used in its self-driving prototype vehicles. The general problem of a varying wireless connection quality in an infrastructure-based cellular network is induced by the movement of the vehicles (Sec. 1.2). To address this problem several Related Works have been published as summarized in the following Section 3.3.

3.3 OPTIMIZED PROVISION USING INFRASTRUCTURE-MANAGED CELLULAR NETWORKS

Through our research study on the current Related Work we could identify three different groups of approaches to improve the general provision of data via infrastructure-managed cellular networks as explained in the following Sections. Namely, they are: i) sharing of measured geo-referenced network performance indicators via a common database, the so called *Connecitivty Map*, ii) online/live estimation of the currently experienced network performance parameters to predict their near future values and iii) decoding of control channel information to retrieve information about the currently active clients and their induced load on the used channel.

3.3.1 *Connectivity Maps*

The fundamental idea behind the Connectivity Map as presented in several publications [73, 75–78, 80] is always to share detailed information about the experienced and measured network quality parameters between the vehicles. Therefore, the Connectivity Map is a central database, where the cellular network measurements are collected from the various vehicles. In the following, this sensor data is aggregated and post-processed to retrieve meaningful information about the overall network performance. The crowd sourced data stored in the Connectivity Map then is distributed to the vehicles and enables them to plan their data transmissions accordingly to the indicated quality of the network along their traveling path [77, 80, 90]. Areas with a good cellular network infrastructure for example can be used to retrieve large amounts of data e.g. over-the-air system updates or the transmission of vehicular sensor data. In areas of rather poor network performance the vehicles can keep the available networks resources free for important HD Map updates and further safety traffic messages. An active maintenance of the Connectivity Map through continuous measurements by the vehicles is considered necessary by the investigated Related Work. The available information from the network operators regarding their cellular networks capabilities (so called network coverage maps - see for example Figure 11 and Section A.5 for further examples) is stated to be often too imprecise. Network coverage maps are most commonly based on mathematical models - not real world measurements. Furthermore they are not comprehensive compared to the Connectivity Map, as they only provide information regarding the availability or absence of a certain network technology.

Geographic influence factors

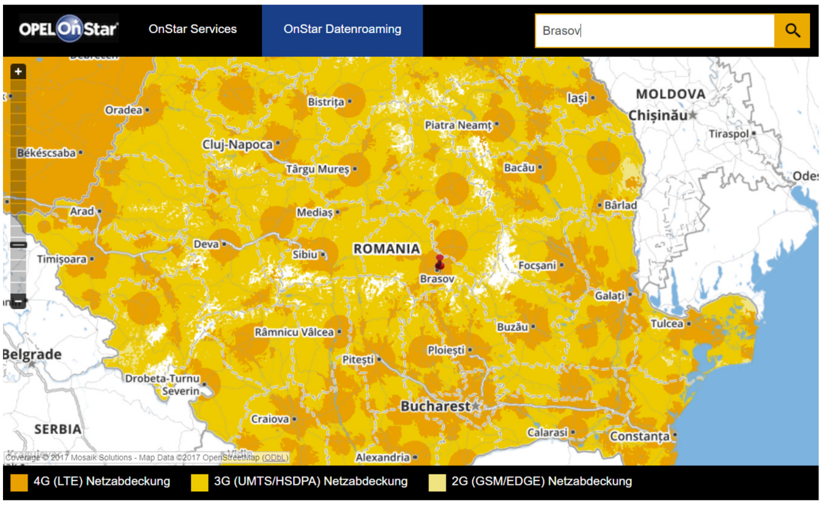


Figure 11: Network coverage map of the Opel OnStar Cellular Service in Romania. The visualized circles of cellular coverage areas indicate mathematical models to be used to generate the map - not actual real world conditions. ©Opel Automobile GmbH[12], Mosaik Solutions, OpenStreetMap Contributors

The majority of the investigated Related Works focused on the possible applications that can be realized with an existing Connectivity Map's information. The addressed optimization of the vehicle's data transmission for example can be achieved through the indication of geographic areas where the vehicle had to perform a handover between the different available generations of network technologies [76] or through the marking of areas with measured high average throughput values [73, 75, 78]. We investigated further Related Work to identify possible concepts to optimize the initial generation and further ongoing maintenance of the Connectivity Map. Several of the identified works focused on an initial efficient distribution of otherwise stationary network nodes [147, 148] in terms of costs, energy consumption and environmental conditions. Approaches that researched mobile nodes [149, 150] to distribute their personal measurements assumed to have the full control over them in terms of driving course, direction and speed. All these requirements cannot be fulfilled in our scenario of self-driving vehicles, as they are given by the passenger's travel destination or external factors such as the present traffic density. There exist various different key performance indicators that describe the overall quality of the cellular network (see Section A.3 and our own contributions in Section 6.1.4, e.g. Table 9). To unleash its full potential as many of those key performance indicators as possible should be stored in the Connectivity Map. Some of the indicators can be obtained free of charge through interfaces of the communication hardware of the vehicles, e.g. the Reference Signal Received Power (RSRP). Others can only be measured by transmitting actual data packets over the network, which is directly related to cellular transmission costs. This for example includes the latency of the connection or the achievable upload and download throughput in the network.

---

[12] https://www.opel.at/onstar/onstar-verfuegbarkeit.html (Last accessed on August 1, 2019)

Concept	Advantages	Drawbacks
PASSIVE	· no additional data traffic required	· no triggering of measurements possible · application bound bandwidth estimation
ACTIVE	· explicit triggering of measurements · peak bandwidth estimation	· additional data for probing / transmission costs / network load

Table 1: Comparison of the advantages and drawbacks of common active and passive throughput estimation concepts based on our investigated Related Work.

To obtain this second kind of key performance indicators, there exist active and passive measuring concepts, which we introduce in the following Section 3.3.2.

3.3.2 *Passive and Active Measuring Approaches*

Application data or ...

The passive measuring approach as for example used in [151, 152] does not transmit data on its own to probe the network. Instead it relies upon other applications, which generate network traffic and thus saves additional transmission costs. The passive measuring approach then calculates the various performance indicators, such as latency and throughput, based on the incoming and outgoing traffic pattern of these other applications (e.g. over-the-air system or HD Map updates, audio and video streams for the passengers, ...). However, the measuring results obtained by the passive approach don't necessarily have to be equal to the cellular networks real capabilities. Most often a certain application, e.g. a video stream, requires a certain network bandwidth, which is, under normal or good network conditions more than fulfilled. In such situations, the passive measuring approach can only indicate that the applications bandwidth requirements are met by the cellular network. Furthermore, is the passive measuring approach dependent on the data flow of other applications. If no data is transmitted over the network, no measurements can be executed. In opposition to the passive measuring approach stands the active measuring as used for example in [73, 75, 78, 80, 153]. By transmitting its own data patterns the active approach probes the cellular network more accurately. This active probing of the network most commonly is realized by the transmission of additional randomized data packets, with no further purpose (so called dummy data). This consequently leads to additional transmission costs. Furthermore important the additional data introduces load on the cellular network, which could otherwise be used to serve the end customer's demands instead.

... random dummy data

Table 1 summarizes the advantages and disadvantages of both techniques.

A very interesting approach to combine both of them together is proposed by Papageorge et al. [154]. Their Measurement Manager Protocol (MGRP) piggy backs existing application data upon cellular probing packets. That way it achieves the same accuracy as the active measuring concepts by introducing way less data overhead

Concept	Advantages	Drawbacks
Connectivity Map	· transmission planning for the whole track of the vehicle	· solely based on historic measurement data
Instantaneous Estimation	· precise estimation of short-term network quality via continuous measurements · approaches often rely on specific hardware / software / provider internal information.	· geographic features not considered by Related Work

Table 2: Comparison of the advantages and drawbacks of the two network quality estimation techniques Connectivity Map and Instantaneous Prediction based on our investigated Related Work.

into the network, similar to the passive approaches. Besides the spatially distributed influence factors, which can be identified via the Connectivity Map, there also exist temporal ones affecting the cellular network's quality as introduced in Section 1.2.

To anticipate these short term, temporal changes in the cellular network exists another technical concept, the so called online or instantaneous estimation of the network performance indicators as described in the following Section 3.3.3.

3.3.3 *Online Estimation*

Temporal influence factors

Online estimation approaches [64, 66–72, 74, 155] rely upon the communication equipment (e.g. the LTE Modem and the connected antennas) installed inside of the mobile device (e.g. a vehicle) to measure the instantaneously experienced network quality parameters. With a small set of collected samples (e.g. over the last few seconds) the different approaches then are able to estimate the future development of the network quality (again for a short period of time in the range of only some seconds to minutes). In consequence the online estimation concept can anticipate changes of the cellular network's quality (e.g. the increase or decrease of the experienced signal strength along the traveled path) quite precisely within a short time horizon (see Table 2). Commenly the proposed estimation concepts rely on machine learning techniques for classification or regression and can be applied to each of the measurable performance indicators of the network. Due to its overall importance for a reliable transmission the throughput of the connection thereby is one of the most frequently estimated performance indicators. Consequently we investigate it as example value in our personal work (Sec. 6.4), too.

The amount of currently active clients in a serving cellular base station (see Figure 2b) cannot be accurately obtained through the two previous mentioned concepts

(Connectivity Map and Online Estimation). Their impact and activity on the cellular networks current capacity is an additional very important indicator that can be obtained from control channel information of the network as described in the following Section 3.3.4.

3.3.4 *User Density Estimation*

User activity as influence on shared medium

The cellular providers know the exact numbers of currently participating clients for each of their cell tower. However, this information is highly unlikely to be shared with third parties due to business confidentiality. For the LTE network infrastructure there exist specialized and costly custom sensing hardware [84]. These hardware kits are able to decode the control channel information provided by the currently serving LTE cell tower to obtain detailed information regarding its capacity distribution between all the active clients. Fortunately with the recent advancements in Software Defined Radios (SDRs) the required hardware setup to realize this decoding functionality became much cheaper and enabled the development of custom software that facilitates similar decoding capabilities at only the fraction of the costs [84, 156–158]. Through our investigation of Related Work in this domain we are aware of at least two open source implementations: "LTE Eye" [156] and "Imdea OWL" [65, 84].

Not all possible communication scenarios can be evaluated through real world tests (e.g. due to cost or time constraints). Such scenarios are instead simulated with specialized software tool-kits as presented in the following Section 3.3.5.

3.3.5 *Further Application In Simulations*

Currently there exists a large variety of simulation software e.g. [159–162], which provide detailed models for vehicular communication scenarios. This includes various plugins to extend the capabilities of the basic simulation software into various further directions. For example the integration of vehicular movement models [106, 163], cellular based communication [107, 164–166] and vehicular focused ad hoc communication [167–169].

3.4 SUMMARY AND IDENTIFIED RESEARCH GAPS

We previously discussed the Related Work accordingly to the three main building blocks of the HD Map maintenance cycle: Generation, Distribution and Provision. Accordingly, we briefly summarize our key findings in the following and highlight the research gaps, which we address in our work.

Distribution of Digital Navigation Map Updates

To solve the disadvantages of common offline and online navigation systems the concepts of incremental and partial map updates have been presented in Section 3.1. This included Related Work to maintain the consistency of the maps after an update, as well as the wireless transmission of the map tiles from a central server to a requesting client (e.g. vehicle).

We consider all of the mentioned scientific contributions as a valuable foundation for our personal work, as these concepts can be applied to any kind of navigation map. However, the identified Related Work mostly focused on navigation systems for manual driven vehicles. Only in the work of Bastiaensen et al. [81] self-driving cars were mentioned as an expected use case, but not investigated with a specific focus.

In comparison to maps, which were created for human drivers, HD Maps have a much higher density and accuracy of geo-referenced static and semi-dynamic information specifically designed for the application in self-driving vehicles. Self-driving cars rely on the HD Map to drive safely and comfortable in otherwise difficult traffic situations, which a human driver could possibly anticipate and avoid without a map (e.g. driving through a construction site). Furthermore the self-driving car always requires a given route or otherwise has to follow along a most probable path (e.g. the current street on which the vehicle is driving) based on the map data to perform the driving task. Consequently we argue that these new preconditions and requirements justify further research regarding new specific and more efficient map update concepts, to keep the data foot print of HD Map updates on a similar level compared to those for navigation maps for human drivers. Based on the related concepts and the requirements of the self-driving vehicles we develop a protocol that provides context-specific map updates individually for each vehicle to further reduce the data transmission footprint and along with it the costs of the update process.

Furthermore, we presented Related Works, which rely upon ad hoc communication technology to share information directly between the vehicles. Many of the investigated related publications focused on highly individual data streams such as video streams to be shared between the cars, often with the assistance of additional Road Side Unites. None of the investigated Related Works tested the application of ad hoc communication on map data, a research gap which we address by our personal work. As map data is relevant for all vehicles in the same geographic area we consider it as a very suitable use case for the application of ad hoc communication.

Aggregation of Vehicular Sensor Data for Map Maintenance

In Section 3.2 we presented different approaches to generate or maintain map data based on collected vehicular sensor data from participating vehicles (Floating Car Data). Most common data source were GNSS traces of the vehicles, which then were combined with further sensor readings such as accelerometer, camera, gyroscope or radar data. Several of the discussed approaches thereby relied upon expensive sensor equipment (e.g. [143]), which is not commonly available and hinders a large scale deployment of the proposed concepts. The majority of the investigated Related Work focused on the generation or maintenance of map data and a high accuracy in the achieved results, which in consequence can often take a long time to process [142]. However a much more critical part in the maintenance cycle of the HD Map is the detection of faulty map data to inform affected self-driving vehicles to take appropriate actions, e.g. to handover the driving control back to their human passenger. We argue that for the detection of faulty information in a given reference map not the same sensor accuracy is required compared to a full correction of this information. Instead the reaction time to identify the changes is the most crucial metric.

As our contribution we propose and evaluate a detection algorithm of changes in the road infrastructure to mark erroneous lane segments in the map accordingly. Therefore the algorithm relies on sensor input from common and ubiquitously available, low-cost sensor equipment, which is not only available in vehicles but also further mobile devices (e.g. smartphones).

Optimized Wireless Provision

In Section 3.3 we presented Related Work to optimize the data transmission of self-driving vehicles in infrastructure-managed cellular networks. All of the discussed approaches have in common to provide information about the cellular network's performance and capacity to optimize the planning process of the data scheduler inside the vehicles. Individually they target different aspects, which are influencing the transmission process. Connectivity Maps provide geo-referenced network quality measurements to plan the long-term data transmission. They address geographic influence factors such as buildings and terrain. The majority of investigated Related Work focused on the benefits that could be obtained from a given Connectivity Map. The initial generation and especially the further maintenance of the information in the Connectivity Map were not in the researchers focus. Existing active and passive measurement approaches to be used for the maintenance of the Connecitivty Map either induce additional unwanted load on the cellular network during the probing process or do not necessarly measure the network's performance to the full possible extend. A very interesting approach presented by Papageorge et al. [154] requires an active stream of application data to be piggybacked upon measurement packets to reduce the additionally created load on the cellular network. If no data is available their concept degrades to active probing.

In our personal contribution we address this problem through our own measuring framework, which intelligently collects the vehicular sensor data and distributes the map updates to actively probe the cellular data link, without using any dummy data.

Instantaneous estimations of the network focus more on information relevant for the short-term data transmission affected by timely influence factors such as weather conditions. Therefore the proposed approaches heavily rely on machine learning algorithms to enable the estimation. The identified Related Works do not take the geographical influence factors into major consideration to adapt their training process accordingly to them [64, 66–72, 74, 155]. Furthermore they often rely on special hardware or software [68–70, 74] to obtain the training set. We propose a new technique to use geographically specific training data to optimize the machine-learning based estimation process and furthermore rely on ubiquitous available sensor equipment for its collection.

Through decoding of the cellular control channel the influence of separate clients on the shared cellular medium can be obtained. To the best of our knowledge all publicly available tool-kits have not been evaluated in a moving environment (e.g. a vehicle), as the continuous changing network parameters results in further requirements for the decoding process. In our contribution we combine network indicators such as the currently relevant transmission frequency obtained from a smartphone to optimize the decoding process and evaluated possible influence factors.

Furthermore we presented simulation frameworks for vehicular communication scenarios and use cases that could not sufficiently be tested in real world setups, e.g. due to cost reasons. Most of the discussed simulation tool-kits have a steep learning curve and require significant computation power, while being limited in the size of their scenarios. In our personal contributions we provide frameworks to easily setup and simulate vehicular communication scenarios to address this research gap.

Through the discussed approaches and identified open challenges in the Related Work we derived our own contributions presented in the following to: i) distribute updates for HD Maps of self-driving vehicles (Chapter 4), ii) detect changes in the street network to mark outdated map data requiring updates (Chapter 5) and iii) obtain broad network performance information to enable a reliable data exchange between the self-driving vehicles themselves and data processing entities via wireless communication (Chapter 6).

4

MAP UPDATES FOR HIGHLY AUTOMATED VEHICLES

In this Chapter we address our first research challenge (see Section 1.2): the reliable and data-efficient Distribution of HD Map Data Updates. Through our investigated Related Work we identified a research gap in the existing concepts for the distribution of map updates. Most of the related approaches focused on digital navigation maps in general, but did not address the particular requirements of navigation systems used in highly automated vehicles (HAV). The Dynamic Map Update protocol as presented in the following Section 4.1 addresses this gap. In it's initial design and evaluation we assume a cellular network as communication medium. In Section 4.2 we extend the protocol to incorporate WLAN based ad hoc communication as well and evaluate the additional performance gains.

4.1 DYNAMIC MAP UPDATE PROTOCOL

We designed, implemented and evaluated the Dynamic Map Update protocol for the requirements and prerequisites of a self-driving vehicle to realize data efficient map updates for HAV navigation systems. This includes a much higher frequency of map updates as well as an initially given route to start the highly automated driving task. Nonetheless the protocol can be also applied to map data of conventional navigation systems, if these systems are actively used by the human driver (a selected route has to be present at the beginning of the trip or otherwise a most probable path to follow along has to be assumed).

4.1.1 *Basic Example of Map Updates via the protocol*

We commence our overview about the protocol with an example scenario, illustrated by Figure 12, to highlight its fundamental properties.

In the scenario our considered map database is represented by six individual map tiles (each spanning some square kilometers of area), which are updated independently from each other. The map tiles currently stored in the internal memory of the considered self-driving vehicle are indicated by grey color. The active route of the vehicle (dotted line) spans from the "Start" to the "Destination" location along the present road network. To navigate to its destination, based on up to date map data, the car exchanges information with a central map data server in the backend through a wireless, cellular connection as explained in depth in Section 4.1.3.

In the example the central map server hosts an updated version of the map with two different map tiles containing two affected road segments (black color).

Via the related map update approaches presented in Section 3.1.1 both of these map tiles would now be provided by the server as partial and incremental map updates to

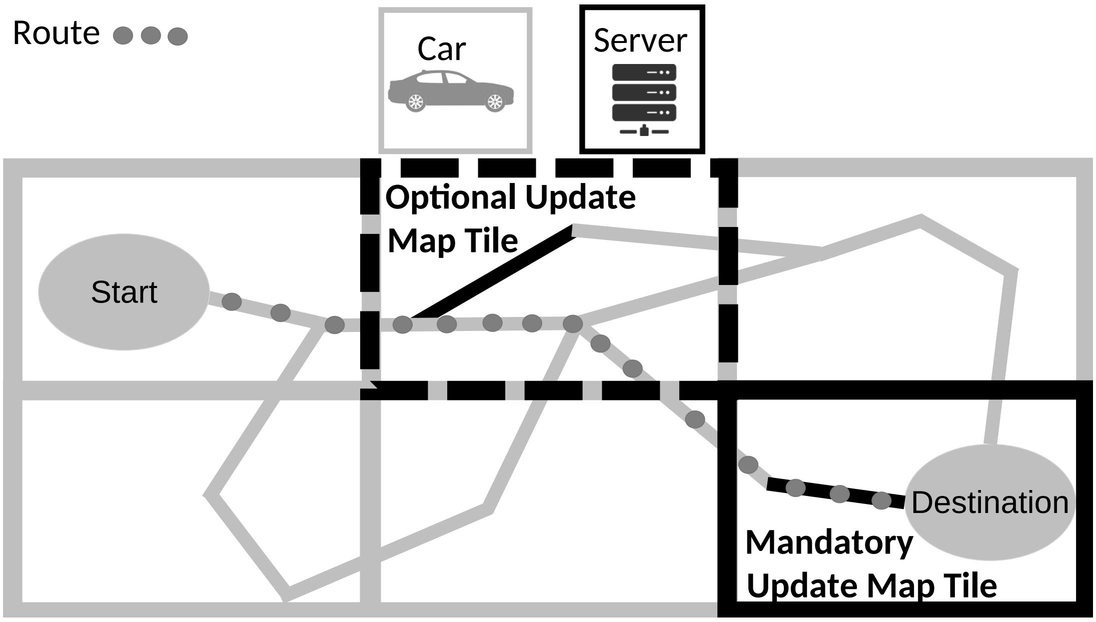


Figure 12: Example scenario for the general working principle of the Dynamic Map Update Protocol. The current version of the map stored inside the vehicle is marked by grey color. Map updates provided by the central server are colored in black [170].

the car. Most of the presented approaches do only identify outdated map tiles in the database and provide updates for them, even if they are not crossed by the current route of the vehicle.

Considering a street map of Germany this could for example include map updates of the area of Berlin, even though the end customer's car is only driving in the area around the city of Cologne.

A slightly more advanced update concept was proposed by Bastiaensen et al. [81] as result of the ActMap project. Their update algorithm only provides map updates for the tiles that are traversed by the vehicle on its trip. However in our considered example scenario this would still lead to the provisioning of both map tiles as the condition is also fulfilled.

Mandatory and optional map updates

As novel contribution our Dynamic Map Update protocol instead first identifies the relevance of each individual map update for the currently selected route before executing further steps. Therefore the protocol distinguishes between mandatory map updates, which directly affect the vehicle on its current route, and optional map updates, which do not immanently affect it, but are located in one of the map tiles that is traversed by the car.

This for example could happen at a highway crossing section, where a construction side only effects one of the two crossing highways with the vehicle only roaming on the other one, staying completely unaffected.

As result the Dynamic Map Update protocol only issues the transmission of the map tile flagged as mandatory in our example scenario. The second map tile instead is indicated as optional update by the protocol to the navigation system of the car. Based

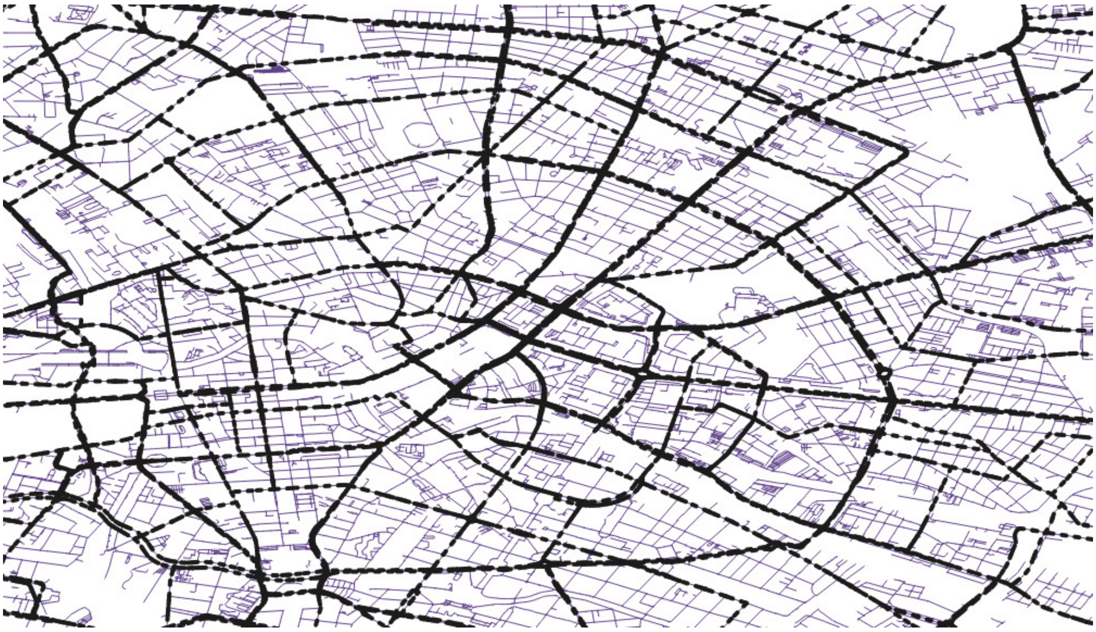


Figure 13: Categorization of streets into highway (thick lines) and city layer (thin lines). The example shows the city center of Berlin [170]. Map data ©OpenStreetMap contributors

on the vehicles configured policy (e.g. to i) use as little cellular data as possible, ii) receive as much updates as possible or iii) receive all updates for frequently traveled areas) the navigation system of the car then can request the transmission of this additional map tile from the server or skip it.

Based upon this motivational example we introduce our fundamental design decisions of the Dynamic Map Update protocol in the following Section 4.1.2. The transmission details of the protocol are then described in Section 4.1.3.

4.1.2 *Fundamental protocol properties*

On the basis of our initial idea to filter the map data accordingly to its relevance for the current route, we developed techniques to integrate this concept already into the map database's storage structure.

Thereby we were inspired by the operation of modern routing algorithms as for example described by Min et al. [24]. These algorithms group the various existing streets into specific subcategories accordingly to their personal type (e.g. motorways and city streets as shown in Figure 13). This hierarchical structuring allows the algorithms to perform route calculations in only a short amount of time.

To reach for example a destination in Berlin from a starting position in Cologne, the algorithms first identify the shortest route to the nearest motorway, as the street of the most relevant category to reach two distant places quickly by car. Then the algorithms only search the network of interconnected motorways to reach Berlin.This saves a significant amount of computation time. The route calculation continuous on

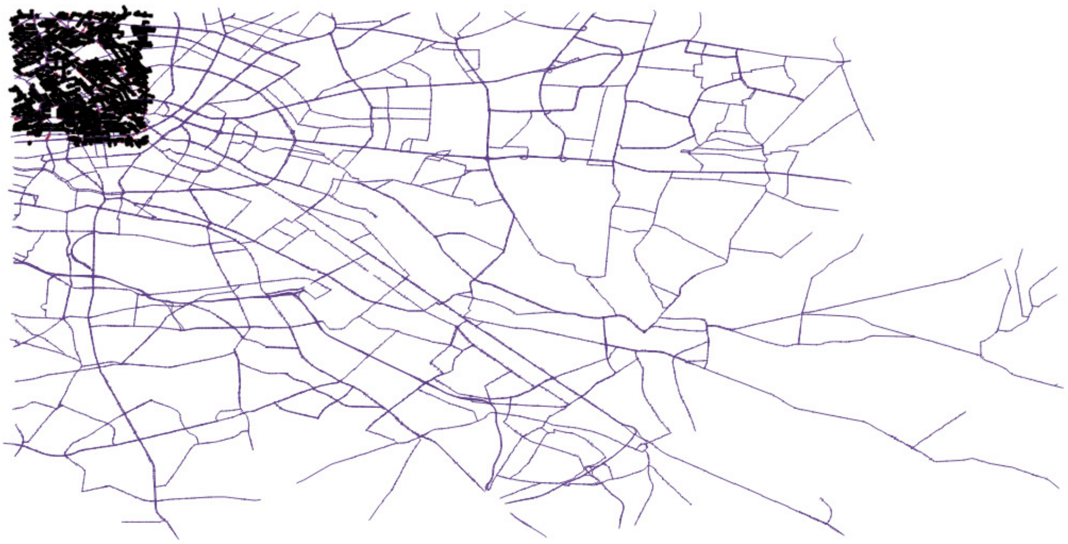


Figure 14: Exemplary visualization of a city street layer map tile (bold lines) in comparison to a highway layer map tile (thin lines). The highway layer map tile reaches form the city center of Berlin in the upper left to the outskirts of Berlin in the lower right [170]. Map data ©OpenStreetMap contributors

the motorway layer until it reaches the outskirts of Berlin. Then urban streets are considered again by the routing process to reach the final destination.

We transferred this hierarchical routing approach into the transmission of map updates.

Different map layers for data-efficient map updates

In contrast to the routing algorithms, which save time in the calculation process, our updating algorithm reduces data that needs to be transmitted.

When the self-driving car requests a route the map update algorithm first investigates which different streets (e.g. highway or city streets) are used to reach the destination position. The algorithm then only replies back to the car with the map tile updates, which are relevant for the related layers of these streets.

For our performance evaluation of the protocol conducted in Section 4.1.5 we subdivided the map data into two different layers (highway and urban/city streets) as shown in Figure 13. A further enhancement of this concept is easily applicable by introducing additional road type categories into the street network.

The network of highway level streets (as visible in the Figure 13, indicated by blue, thin colored lines) is more sparse in comparison to the dense network of city level streets (bold, black lines).

We take this property into account by storing the different road types into map tiles of different sizes as shown in Figure 14.

That way similar amounts of map data are transmitted when processing map tiles of either of the two different layers. Furthermore the control data overhead required by the protocol to index the different map tiles scales more efficiently with the traveling distance covered by the given route.

Geohashes as index

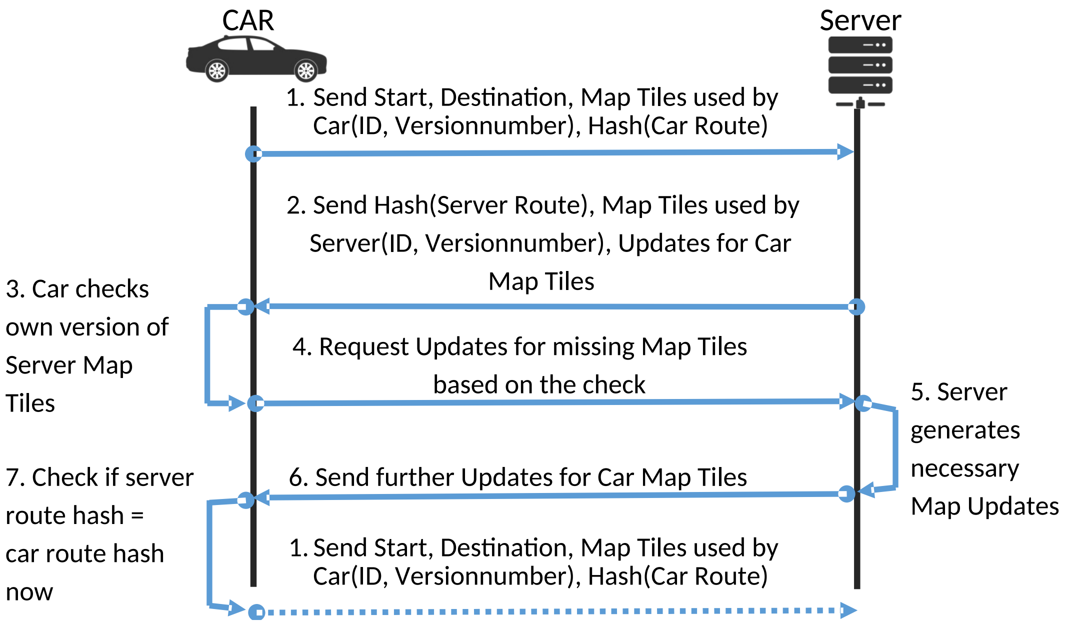

CAR
Server
1. Send Start, Destination, Map Tiles used by Car(ID, Versionnumber), Hash(Car Route)
2. Send Hash(Server Route), Map Tiles used by Server(ID, Versionnumber), Updates for Car Map Tiles
3. Car checks own version of Server Map Tiles
4. Request Updates for missing Map Tiles based on the check
5. Server generates necessary Map Updates
6. Send further Updates for Car Map Tiles
7. Check if server route hash = car route hash now
1. Send Start, Destination, Map Tiles used by Car(ID, Versionnumber), Hash(Car Route)

Figure 15: The map exchange sequence of the Dynamic Map Update protocol [170].

To realize this functionality we rely upon the geographic indexing concept of Geohashes [104] as introduced in Section 2.5.

This allows our updating algorithm to interchange seamlessly between the different layers of map tiles in various sizes, when following the given route for the identification of possible updates.

Furthermore the Geohashes enable a possible future extension into more individual map tile layers.

With these fundamental technical prerequisites present, we now designed the details of the data exchange to be executed via the Dynamic Map Update protocol as explained in the following Section 4.1.3.

4.1.3 *Protocol details*

The main purpose of the Dynamic Map Update protocol is to provide mandatory and if requested optional map updates to self-driving vehicles. That way the protocol ensures up to date map data for the vehicle to route on with only a small data footprint for the update procedure, which keeps the transmission costs low. The procedure, which we developed to achieve this goal is realized in a sequence of seven individual steps as illustrated by the diagram of Figure 15. They are explained in the following based on our initial example with further protocol details as shown in Figure 16.

1.) At the beginning of its trip the car starts the map update request. Therefore it sends its start and desired destination position to the server. Based on these two positions the vehicle then performs its personal route calculation using its currently present (not updated) map data. From this calculated route it provides the server the

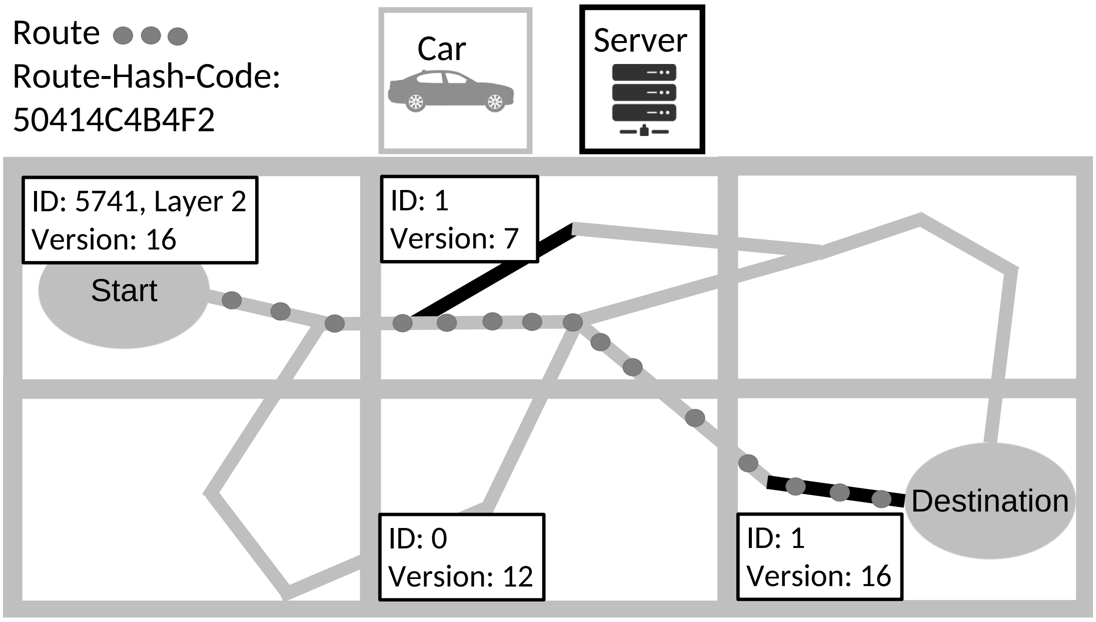


Figure 16: Detailed example for the used sequence of map tiles [170].

information about the map tiles, which it has to traverse. The provisioning of this information happens in a data efficient way. Only the ID of the initial (start) map tile (ID 5741 in the example of Figure 16), as well as its layer type (e.g. 1 for highway and 2 for city streets) is transferred as a whole to the server. The following map tiles are then only identified by one number with regards to their position related to the previous one (ID 0 = south, ID 1 = east, ID 2 = north, ID 3 = west). If the car changes its layer in the map tile this is as well indicated by an additional number. To simplify the drawing we assume that the example vehicle stays on the layer 2 throughout its complete trip (2 for city streets).

Hashcode derived from street segment IDs to identify route

Furthermore the car has to provide the version number of each of the used map tiles to the server (e.g. version 16 for the starting position's map tile). As last initial parameter the vehicle sends a unique hash code (e.g. 50414C4B4F2) to the server, which is generated based on the exact route, which the car calculated. Therefore each individual street segment is marked by a unique version ID (based on its location in the map and the server-time this segment was last updated), which serve as possible input to the correlated hash function.

2.) After having received all the information from step 1.) on the server side, the server will start to calculate a route from the start to the destination position based on its own up to date map data to avoid routing into traffic hazards. From this route it also generates the correlating hash code. The server then compares its own hash code with the one of the vehicle. If both hash codes do not match, the vehicle needs to receive immediately mandatory map updates, which are affecting its current route. This process is explained in the subsequent steps 3.) till 6.).

If both codes match it means that at least for the current route the map of the vehicle and the information in the map database of the server match. If this is the case the server continues to identify possible optional map updates, which the car might consider for an update, too. Therefore the server compares the version numbers of the map tiles provided by the car with its own. If optional map updates are present the server will send the related map tile IDs to the vehicle. Based on the car's policy (e.g. driving criteria or remaining cellular data volume) it requests or skips these further updates from the server.

A furthermore possible case is that the map server uses completely different map tiles than the vehicle to reach the selected destination. If this situation appears the server provides the vehicle with its own route information. This also includes the hash code for the calculated route, as well as the IDs and version numbers of all crossed map tiles. This last step is necessary to ensure that the car does not find another as well outdated alternative route in its own database after having received the mandatory map updates.

3.) till 6.) In the following steps the car then has to finalize the updating processing by comparing the information received form the server with its own database. It checks the indicated map tiles of the server and if necessary requests the additional required mandatory map updates. These as well as further requested optional map updates are then transferred from the server to the car.

7.) In few occasions the provided map updates might not ensure an up to date route calculation on the cars side. The vehicle still might find another outdated alternative route after completing all map updates. The protocol however has to ensure that car and server calculate the same route in the end. Consequently as a last step the car has to recalculate its route with the updated map data and then again compare its new hash code with the one of the server. If both codes still do not match, the process has to be repeated with the new preconditions at hand.

Besides this update process we assume the map server to provide immediately further map updates to the requesting vehicle, if new ones occur during the current trip of the car.

After this in depth explanation of the working principle of the Dynamic Map Update protocol, we conducted a performance evaluation of the protocol by simulating different scenarios based on map data of the German city Berlin as described in the following Section 4.1.4. The achieved evaluation results are then described in the subsequent Section 4.1.5.

4.1.4 *Map Update Simulation Scenario - Berlin example*

Evaluation requires sufficient amount of map changes

To correctly evaluate the performance of the Dynamic Map Update protocol we require an example map database with a sufficient amount of historic changes. Several different map formats have been developed for the application in HD Maps. This includes for example the specifications of ″OpenDrive″ [171], the ″Navigation Data Standard - NDS″ [172] and ″lanelet″ [173].

To the best of our knowledge there are currently (2019) only minor sample maps of these formats[13,14] available to showcase the functionality and the properties of the map standards. In consequence they are not suited for our evaluation.

Due to these preconditions we instead decided to rely upon map data of the community driven, open source project ″OpenStreetMap″ (OSM) [174] for the evaluation.

Besides having a strong community driven database with daily conducted map updates, the OpenStreetMap map data possesses several further properties that supported our evaluation task. The project for example makes the map data available as database dumps at time intervals of one minute, one hour or one day, between the most current material and its predecessor. It also categorizes the different streets (motorway, trunk, primary, secondary, tertiary, ...), which allows us to group them in our specified map layers (highway and city streets). Furthermore each street segment is identified via a unique number, which allows the application of the hash function for route calculation.

For our evaluation we chose a Geohash string length of four letters for the highway street layer and five for the remaining streets on the city layer level. Thus the Geohashes correlate with bounding box sizes of $39.1x19.5km^2$ and $4.89x4.89km^2$ respectively. Consequently a highway layer map tile covers the same area as 24 city street layer map tiles (see Fig. 14).

To resemble the application scenario of an HD Map as closely as possible, we selected the OSM map data of the German city Berlin and its surrounding regions (see Figure 89) for our evaluation process. This region of the OpenStreetMap receives frequently a high number of updates, as the community of volunteer map maintainers is especially strong in this area.

Comparison of database dumps of 1 and 15 days

In our evaluation we compared the databases dumps of consecutive days, as well as a period of two weeks in between each other to mimic the frequent changes of HD Maps, which they likely achieve in far less time. To ensure the significance of these tests we collected data sets of 30 consecutive days of OpenStreetMap database dumps (1st till 31st August 2016). From these database dumps we then extracted changes of the map data in between the different dates with periods of 1 and 15 days time difference to simulate situations with a smaller and a larger amount of map changes.

Although we developed the Dynamic Map Update Protocol with HD Maps in mind, the protocol itself can be applied on navigation map data with any given resolution. The absolute amount of data to be saved will only scale accordingly to the amount of introduced updates. Thus in consequence the relative relations between our chosen reference algorithm [81] and the Dynamic Map Update protocol, as explained in the following Section 4.1.5 are of higher importance than the actual measured numbers.

---

[13] http://www.opendrive.org (Last accessed on August 1, 2019)
[14] http://www.openlanemodel.org (Last accessed on August 1, 2019)

4.1.5 *Dynamic Map Update Protocol - Evaluation*

For the performance comparison of the Dynamic Map Update protocol we selected the concept of route calculation specified by Bastiaensen et al. [81] as our reference algorithm. In the following we refer to Bastiaensen's approach as simple map update algorithm, as it triggers the transmission of all incremental map tile updates along the current route of the vehicle. The Dynamic Map Update protocol in contrast shall reduce this amount of data by identifying the mandatory and optional map updates as described in Section 4.1.1 ff.

Different trip lengths according to daily driving behavior

A study by Papageorge et al. [154] has shown that the average daily driving trip length of a European is between 10 to 30 kilometers. We take these findings into account for our evaluation by configuring 60 % of all our generated trips to be within this distance range. The remaining 40% of the trips are divided up into half of them being below and the other half being above this range. To generate the trips we randomly selected start and destination points and identified the interconnecting route using the Dijkstra algorithm.

To evaluate the performance of the Dynamic Map Update protocol in comparison to the simple map update approach we selected two different metrics as discussed in the following two sections. On the one side we compare the amount of map tiles, which have to be processed on the server side to generate the incremental map updates for the requesting vehicles (the server load). On the other side we compare the amount of actual map objects that need to be transferred via the cellular connection from the server to the requesting car (cellular data - costs). An excerpt of the detailed evaluation results is shown in Table 3 for two simulation runs of consecutive days.

Load savings on server side

To evaluate the individual processing load induced by both protocols on the map providing server we generated 10.000 individual trips as input for the evaluation. For this set of trips and the map data of one day of updates, the simple map update algorithm required to process a median of 18226.5 map tiles to identify all updates (see Figure 17). The Dynamic Map Update protocol, which only provides the mandatory map updates, needed to process a median of 807.5 map tiles. This is less than 5% of the data of the reference algorithm. As expected with an increase in the considered time difference (15 days) and consequently the amount of overall updates in the map tiles this effect was slightly reduced. Still the Dynamic Map Update protocol only required to process a median of 22% (9,798 map tiles) of the data that was investigated by the simple map update approach (43,968.5 map tiles) to generate the incremental updates.

Data savings on client side

On the side of the vehicular client the amount of received data, which stands in direct correlation to the transmission costs is considered by us as the most important performance metric. To investigate these parameters we created an additional scenario, which consists out of 100 cars. Each of these vehicles individually requested map

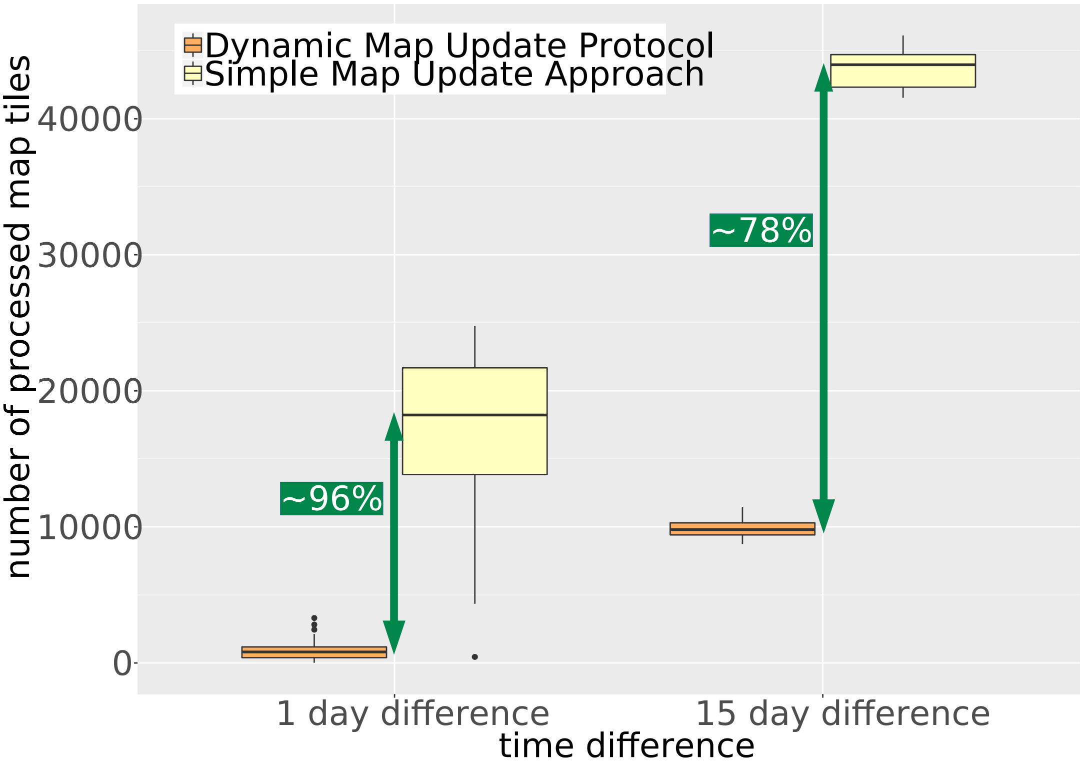


Figure 17: Number of Map Tiles which have to be processed for transmission for 10,000 independent trip requests [170].

information for 100 consecutive trips. Thereby we assumed that the map information from each preceding trip request is stored in the internal storage of the vehicle and thus is available in the following. This resembles situations where for example important interconnecting routes are often used and thus do not need to be updated over and over again. We selected the amount of 100 consecutive trips to show possible saturation effects in the map updating process. This is achieved, as the considered scenario of 100 trips in a temporal sequence only leaves around 5 minutes for the completion of a trip, assuming 8 hours of drive time (e.g. a taxi driver). The obtained evaluation results are shown in Figure 18 for map changes of 1 day time difference and in Figure 19 for 15 days time difference, before the update is executed. Both series of boxplots (1 and 15 days) clearly show that the Dynamic Map Update protocol in average always had to provide far less map objects to the requesting vehicle, in comparison to the simple map update approach. With reference to Papageorge [154] we expect that the majority of drivers perform far less trip requests on average. For this situation the Dynamic Map Update protocol is even more beneficial, as its provided amount of map objects increases much more steadily in comparison to the simple map update approach when requesting further trips. To verify the statistical significance of our evaluation, we furthermore conducted a paired t-test on the obtained results. Therefore we specified the following

Table 3: Exemplary excerpt of the achieved test results [170].

Execution Run	Old Map Date	New Map Date	Dynamic Map Update Protocol Tile Updates	Simple Map Update Approach Tile Updates	Tile Savings in %	Dynamic Map Update Protocol Objects with Changes	Simple Map Update Approach Objects with Changes	Object Savings in %
1	1-8-2016	2-8-16	1,128	7,413	84.78%	30,427	85,734	64.51%
2	2-8-2016	3-8-16	169	3,252	94.80%	597	9,567	93.76%
...	...	...	...	...	...	...	...	...

Table 4: Obtained p-values of the paired t-test for the results of the second test [170].

number of requests	10	20	30	40	50	60	70	80	90	100
1 day time difference	1.90e-06	1.46e-06	1.06e-06	2.69e-06	9.75e-07	5.40e-07	1.40e-06	1.42e-06	1.36e-06	1.40e-06
15 days time difference	8.60e-09	3.96e-09	1.36e-09	7.14e-11	1.39e-12	1.69e-13	3.43e-11	2.12e-10	4.14e-11	8.03e-11

two hypotheses. The null hypotheses H_0: "The two map update approaches distribute the same amount of update data."

$$H_0 : \mu_{dynamicMap} = \mu_{simple} \tag{1}$$

As well as the alternative hypothesis H_1: "The Dynamic Map Update protocol transmits less data than the simple map update approach."

$$H_1 : \mu_{dynamicMap} < \mu_{simple} \tag{2}$$

Where μ is the mean of the number of updated map objects.

Table 4 shows the obtained p-values for the series of consecutive requests. These p-values clearly indicate that we can reject our initially stated null hypotheses H_0 and accept the alternative hypothesis H_1.

To verify that the selection of 10 cars and their consecutive trips was not in favor of the Dynamic Map Update protocol, we conducted a further evaluation, where we let 1000 independent vehicles perform 10 consecutive trips. We expect this scenario to be more closely to an actual real world usage scenario. The obtained results therefore are

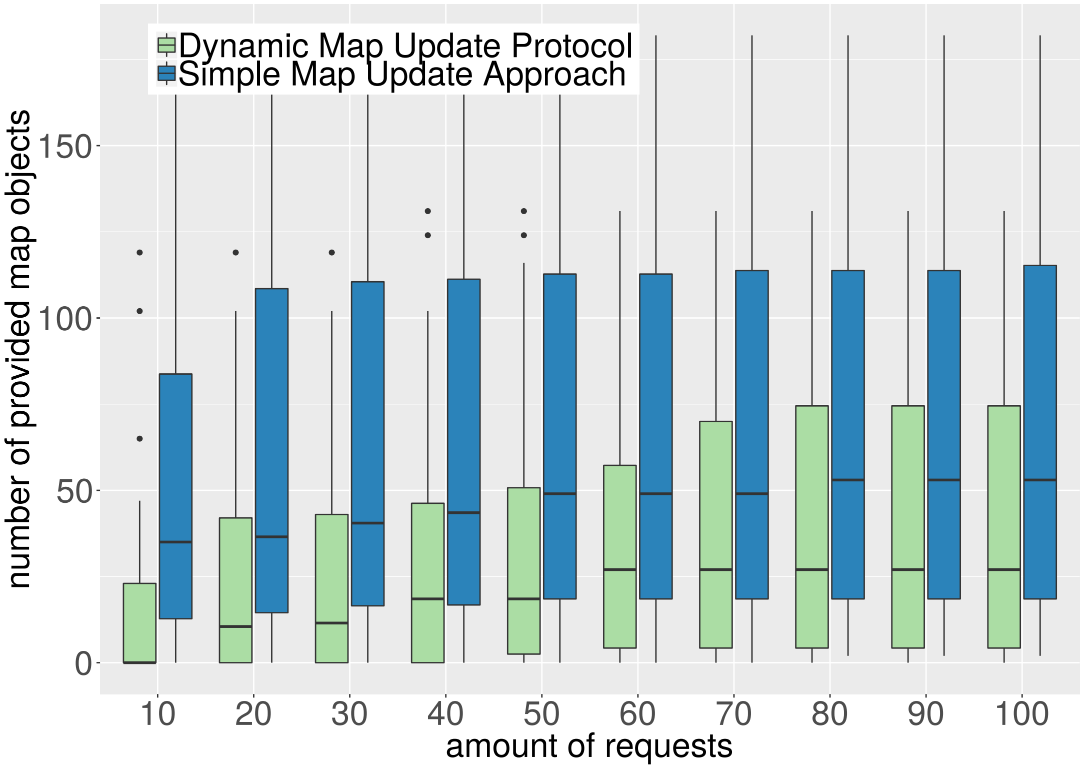


Figure 18: Map objects provided after 10 to 100 consecutive requests. Databases with 1 day difference [170].

shown in Figure 20 and show similar qualitative results. Also for only 10 consecutive trips and a larger variety of vehicles the Dynamic Map Update protocol shows similar data-saving capabilities.

In summary of this evaluation we could showcase the capabilities of our proposed context-selective Dynamic Map Update protocol to reliably and data-efficiently distribute map updates for vehicular based digital navigation systems. In comparison to existing distribution algorithms the protocol significantly reduces the amount of data and workload necessary to realize the updates, which is an especially important criteria in consideration of updates for HD Maps as used by self-driving vehicles.

4.2 HD-WMAP EXTENSION

Ad hoc communication for map data sharing

In the initial design of the Dynamic Map Update protocol we considered the cellular network to realize the wireless data transmission of our map updates. As introduced in Section 3.1.2 however modern ad hoc communication technology enables the cars to exchange directly information with all of their surrounding neighbors in the ongoing traffic as well. We considered the usage of this communication technology as a great possibility to further reduce the costs of the HD Map update distribution process. As result we developed and evaluated the so called HD-Wmap extension, which

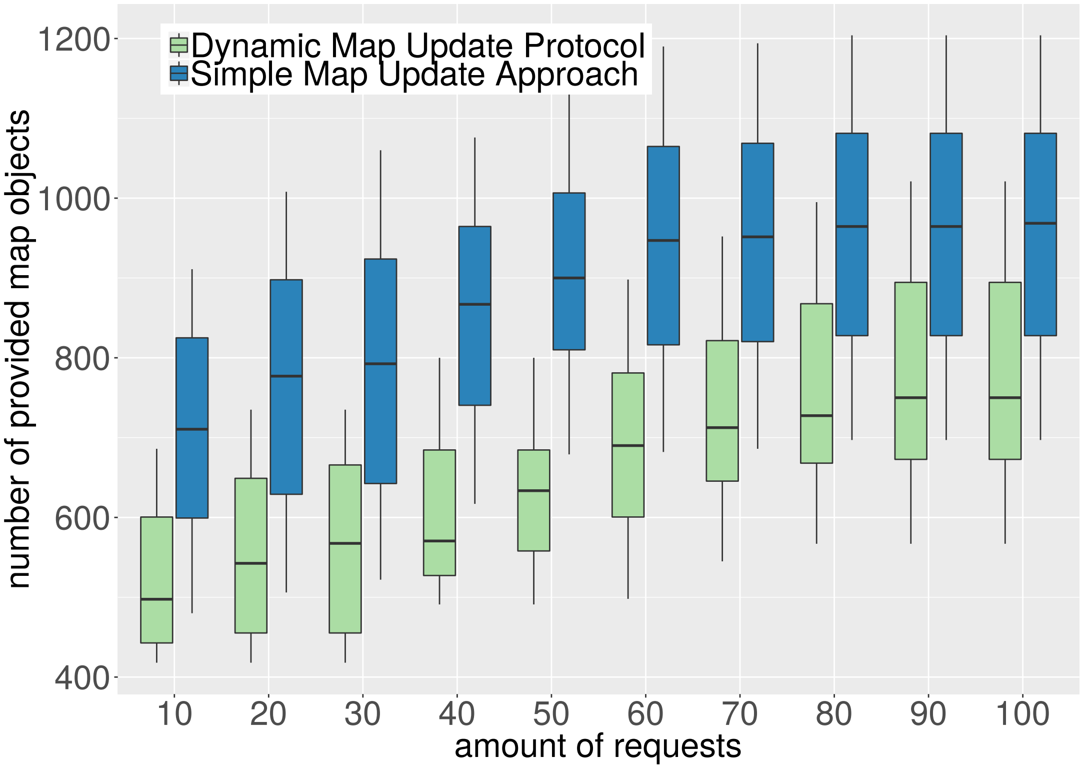


Figure 19: Map objects provided after 10 to 100 consecutive requests. Databases with 15 day difference [170].

Realization via existing standards

incorporates these ad hoc communication features into our existing protocol's process. Our main design decision for the HD-Wmap extension was to realize all features by completely relying on already existing ad hoc communication standards, e.g. in terms of defined message types and transmission technologies. That way we enable an easy and reliable future deployment of our proposed concept. As explained in Section 3.1.2 there currently exist various different physical communication technologies (e.g. IEEE 802.11p and LTE-V2X), which enable the ad hoc communication between adjacent vehicles. The HD-Wmap extension works with any of those technologies. However for our evaluation as shown in the following Section 4.2.3 we are considering the IEEE WLAN standard 802.11p to realize the ad hoc communication between the cars, as well as a cellular connection from the cars to the map server in the backend. That way we are referring to the Related Work of Section 3.1.2 in terms of off-loading data transmissions from the cellular to the WLAN network. In contrast to the Related Work we did not consider dedicated Road Side Units for the performance evaluation of the HD-Wmap extension, as we wanted to evaluate the possible performance gains based purely on the inter-vehicle communication. In our opinion this approach is more likely to be realized in the future due to monetary reasons. However the HD-Wmap extension surely would benefit from a deployed infrastructure of Road Side Units as well. Several of the message types, which have been defined for vehicular ad hoc communication (e.g. CAM, DENM and SAM see Section 3.1.2). include optional information containers

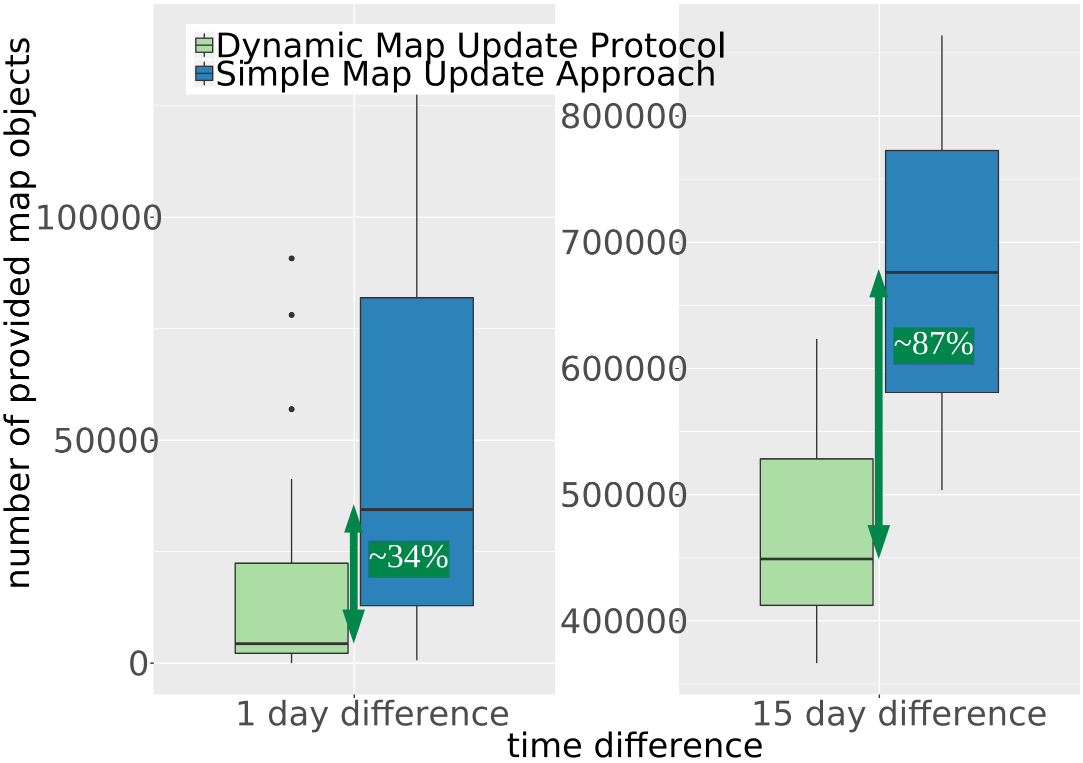


Figure 20: Map objects provided to 1,000 independent vehicles performing 10 consecutive trips [170].

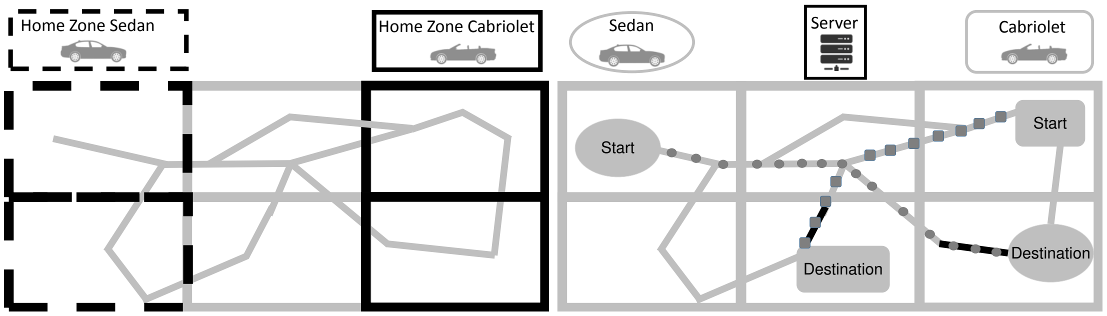
in their specification. We propose to use one of these optional containers to store the additional information required to realize the map data sharing procedure between the vehicles as explained in the following sections. For our evaluation in Section 4.2.3 we assume the required data to be piggybacked upon some of the frequently broadcasted CAM messages (Cooperative Awareness Message) [59] when requesting map updates. That way we ensure a fast distribution of the messages to all neighboring vehicles.

4.2.1 *Home zone concept*

The Dynamic Map Update protocol realizes the concept of mandatory map updates introduced in Section 4.1.1 on a partial and layer-based level. Consequently the vehicles do not necessarily possess the similar map information (all layers of a map tile) as the backend map server. To facilitate the individual sharing of map data between the cars, we introduce the concept of the ”home zone” with the HD-Wmap extension. Under the technical term home zone we consider the map tiles, which are most frequently roamed by each individual vehicle. These map tiles include for example areas such as the local neighborhood, nearby towns or the commuting routes to work. As these map tiles are most relevant for the considered vehicle, we propose that they are frequently updated via a full update of all layers, instead of the common layer-specific update. These full map updates can be scheduled to be executed for example at home when the car is parked in the garage via a WLAN connection. That way the home zone concept realizes the prerequisites for an efficient ad hoc map distribution, as foreign vehicles are more likely to meet a locally driving car when requesting a map update of the upcoming areas.

4.2.2 *Procedure of the HD-Wmap extension*

With regards to the introduction of the Dynamic Map Update protocol in Section 4.1.1 we refer to its map section example to introduce the enhanced functionalities provided by the HD-Wmap extension to the protocol. In the changed scenario as illustrated in the Figures 21a and 21b we are considering not one, but two vehicles (a sedan and a cabriolet), which are independently from each other requesting map updates for their designated routes. In our example scenario both vehicles are assumed to possess a home zone of two map tiles as indicated by Figure 21a. The single protocol steps, which are enabling both cars to retrieve their map updates are summarized by the Algorithm 1. As first protocol steps both vehicles start their route calculation and request possible map updates as described in step 1.) of the common Dynamic Map Update protocol (Sec. 4.1.3). By using the extended protocol the map server now does not directly answer upon those requests with the provisioning of map updates. Instead it only informs the vehicles of their mandatory and optionally required updates by providing the related Geohashes and tile versions to the requesting vehicles. In our example of Figure 21b this includes two map tiles with one mandatory map change for each of the vehicles. First the cars try to gather these indicated updates via ad hoc communication. To increase the chance to identify a suitable vehicle in the surrounding traffic environment the requesting vehicle waits till a certain configurable driving distance to the required map tile is reached (2.a). For our evaluation we assume this to be the width of one map tile, which is reached when the car enters one of the considered map tile's neighbors. After this distance point is reached the vehicle leverages the already periodically send broadcast messages to include the optional information regarding the required map updates. Therefore the car has to provide two identifiers in the optional container. Namely they are the Geohash ID and the required version number of the specific map tile. The self-driving vehicles only can provide full map tile updates to their requesting neighbors, as they cannot (as done by the map server) store the complete history of the map tile in their own internal storage space. In consequence depending on the size of the map tile such an update can be a small number of megabytes [16]. The IEEE 802.11p standard [112] however has been specifically optimized to transmit small amounts of data (several hundreds of bytes). To address this circumstance we propose to split the individual map tiles before their transmission into smaller data chunks, which are then sequentially transmitted to the requesting vehicle. In correlation to the time, in which the sending and the requesting vehicle are in the necessary close proximity of each other, several of those data chunks can be transmitted. This furthermore allows the requesting car to retrieve data chunks from several sources in parallel if more than one vehicle can provide the required map data. If the reception of data chunks fails, the requesting vehicle



(a) Examplary home zones of two cars (sedan, cabriolet) consisting out of two map tiles.

(b) Indicated routes of the cars (circle = sedan / square = carbriolet) and updated map data (black) provided by the server.)

Figure 21: Example scenario to further illustrate the ad hoc sharing of map data via HD-Wmap [88].

sends further broadcast messages to retrieve the outstanding data chunks from other surrounding cars. Only if the requesting vehicle is not able to retrieve all data chunks via ad hoc communication before a certain minimum distance limit is reached it will immediately request the remaining data chunks from the central map server via a cellular transmission. In our evaluation this distance is reached when the vehicle enters the map tile which requires the update information. This so retrieved full map tile update will then be added to the internal storage of the vehicle as well and becomes available for further sharing with the neighboring vehicles (3.)). If no map information at all could be retrieved via ad hoc communication the vehicle requests again the same map information from the map server stating the urgent need. This time the map server generates and provides the required incremental map update to the requesting vehicle. This situation is for example the case of the cabriolet as the sedan cannot provide the required map data as it is not part of the sedan's home zone.

Algorithm 1 : Actions performed in a map exchange via HD-Wmap [88].

```
HD-Wmap procedure actions:
1.) Check which updates are required for the current trip.
   Gather necessary Geohashes and tile versions from the backend server.
2.a) If (request_distance >= vehicle distance to map tile > min_distance)
  request map tile via ad hoc communication.
  2.b) If (vehicle distance to map tile <= min_distance) immediately download
  the required map update via the cellular network.
  2.c) Else wait and do nothing.
3.) Answer requests, if data is available in the own internal storage,
  e.g. recently downloaded or as part of the home zone.
```

Additionally to this baseline car to car sharing concept we thought about further possibilities to improve the overall ad hoc sharing quota of map tiles. As we did not want to introduce further Road Side Units into the process, due to their maintenance costs, we thought of other possibilities to enhance the already existing mobile communication partners. Public transport vehicles have the unique property to roam continuously on the same streets in the same area throughout the entire day. In consequence they are perfect candidates to provide map data to requesting vehicles in their proximity. This lead us to the idea to enhance public transport vehicles such as buses or tram into mobile data beacons, which are already equipped with the complete map data for their full trip area to share it right from the start with the surrounding vehicles. This modification can be realized with reasonable costs, as only the internal storage of such vehicles has to be big enough to contain the additional data at once.

4.2.3 *Evaluation of the HD-Wmap extension*

In contrast to the initial evaluation of the Dynamic Map Update protocol, where we assumed the constant availability of the cellular network, ad hoc communication is affected by the continuous vehicular movement and the resulting connection and disconnection of nearby communication partners. To evaluate the performance of the HD-Wmap extension under these conditions, we relied upon the traffic simulator SUMO, introduced in Section 2.6 to simulate realistic vehicle movement patterns. To be able to investigate the influence of public transport on the sharing quota we changed the simulated scenario from the city of Berlin to the area of Luxembourg. We did so as the Luxembourg SUMO Traffic (LuST) scenario [109] in contrast to the Berlin scenario provides a realistic simulation of daily traffic, that also includes public transport bus lines (indicated by red color in Figure 91).

We relied upon the simulation environment Simonstrator.KOM [175] to connect the SUMO traffic simulation with a realistic network environment realized through the PeerfactSim network simulator [176].

Parameter	Value
Transmission range	300 meter
Size of one complete map tile (assumed to be independent of its layer)	10 megabytes
Data chunks per map tile	30 (330 kByte per data chunk)
Shared ad hoc channel capacity	10 Mbit/s
Simulated amount of vehicles	5803 including 79 buses
Time of day	22 to 24 o'clock
Home zone of vehicles	initial starting map tile for normal vehicles / the complete route for buses, when considered as data beacons

Table 5: Selected configuration parameter for the Luxembourg simulation scenario of the HD-Wmap extension.

To showcase the performance of the HD-Wmap extension even under difficult communication conditions we created a rather restricted communication environment. The selected simulation parameters are summarized by Table 5. This includes the configuration of a conservative circular transmission range of 300 meters for the IEEE 802.11p standard in our city-based environment, as well as a maximum shared channel capacity of 10 Mbit/s. In the Related Work [115] transmissions ranges of up to 1000 meters and 27 Mbit/s are discussed to be reachable with the 802.11p standard. Furthermore we selected the simulation of a night time scenario between 22 to 24 o'clock, where less cars are driving on the streets. Still 5803 vehicles including 79 buses were present on the streets to reach their individual destinations on the map during this time of the day. We conducted two different simulation scenarios to investigate the possible influence when relying on public transport as mobile data beacons. Therefore we compare the ad hoc sharing quotas, which could be reached when all public buses were operating as normal vehicles in comparison to them behaving as data beacons with an extended map data storage. Normal vehicles only were assumed to initially contain the map tile of their starting position in their internal map storage. Each scenario was conducted on sets of 30 days of OpenStreetMap map data from January 2018 to ensure the significance of our obtained results. With an optimization of the mentioned parameters, e.g. an increase in the transmission range, channel capacity or a higher vehicular traffic density to perform map exchanges we expect even better results. The obtained evaluation results for the whole two hours of simulation run time are visualized by Figure 22.

In the Luxembourg scenario the amount of data required to be transmitted via the costly cellular connection could be reduced by 43.6 percent on average using ad hoc communication between the vehicles when considering their initial starting points map tile to be their home zone. The large variance in the box plots however shows that the actual sharing rates highly depend on the selected routes and the updated map data of each day. With favorable map data available the ad hoc sharing quotas could reach up to 80%. The sharing quota achieved by deploying buses as data beacons however did not show a significant difference compared to a scenario, where all roaming vehicles behaved the same (see Table 18 in the Appendix for details). The little influence is likely due to the small amount of buses (79 in total) compared to the number of other vehicles (5,724) in the scenario. Furthermore we assume that with a proceeding simulation also the map caches of the common vehicles are provided quickly with shared and updated map data, which further diminishes the necessity of the extended map caches in the public transport vehicles.

In summary the obtained evaluation results of the Luxembourg simulation scenario showcase the beneficial usage of ad hoc communication to further improve the map data sharing process of the Dynamic Map Update protocol, especially in terms of transmission costs but also workload on the central server. These overall promising positive results were achieved even without the deployment of additional Road Side Units or the enhancement of participating mobile communication nodes (e.g. public

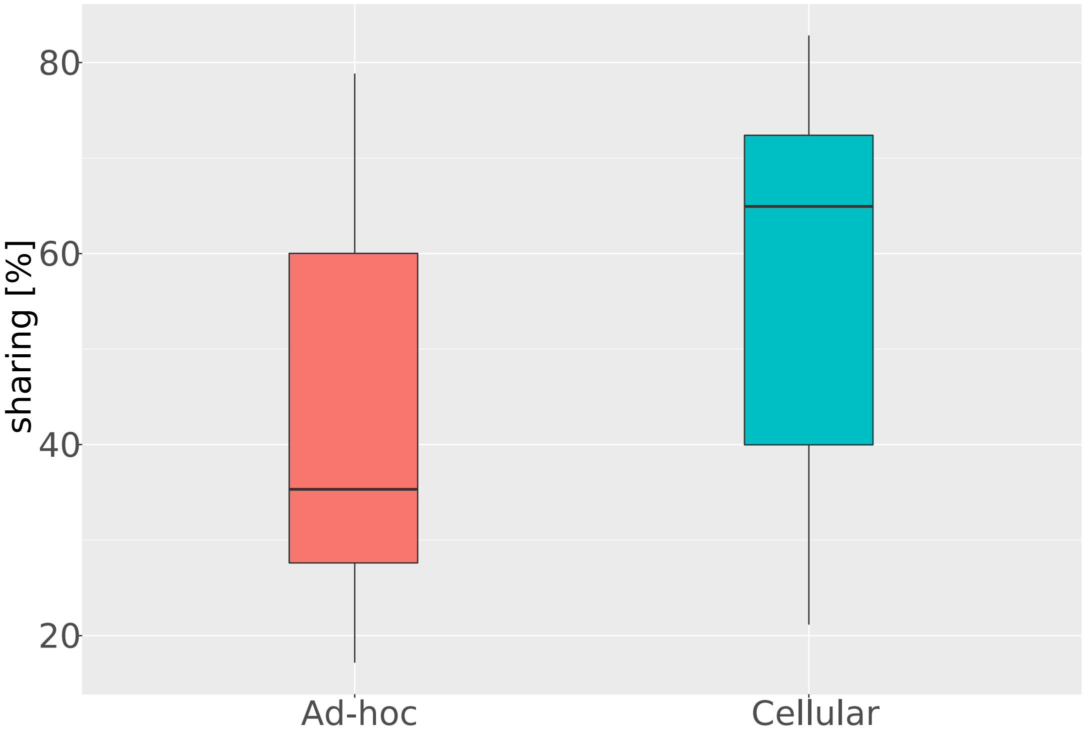


Figure 22: Comparison of the sharing quotas between cellular and ad hoc communication (~64,9% vs ~35,1% in median). All vehicles were initially configured with the map tile of their starting position as already cached map data (their home zone).

transport vehicles).

Besides the data efficient distribution of the HD Map data, as addressed by our presented work of this chapter, also its continuous maintenance opens up new challenging problems as addressed in the following Chapter 5.

5

ROAD INFRASTRUCTURE CHANGE DETECTION

In this Chapter we focus on the second identified research challenge in the maintenance cycle of the HD Map as described in Section 1.2 and illustrated by Figure 1: the fast and reliable detection of road infrastructure changes. The changes have to be identified to maintain the correctness of the related HD Map data and in consequence ensure the safety and comfort of the self-driving vehicles. As our personal contribution in this research field we propose an algorithm to detect changed lane courses (e.g. induced by a broken down vehicle, a construction site and other temporary road obstacles). Through the algorithm related areas in the HD Map are marked as erroneous (to trigger a subsequent update) and affected self-driving vehicles can take appropriate actions (e.g. handover driving control) to avoid otherwise critical driving situations.

Based on our literature analysis we describe the initial idea for our work at the beginning of the following Section 5.1. Subsequently we explain the individual components and conceptual details of our algorithm. To test the algorithm no suitable test data was publicly available. Consequently we generated our own evaluation data set through a measuring campaign on actual German streets and highways as described in Section 5.2. For the data collection we used common smartphones, which were placed on the dashboard inside the vehicle. Based on this data set, which we also made publicly available, the evaluation of the algorithm in Section 5.3 is executed on a real highway scenario with an ongoing construction site.

5.1 DESIGN OF THE LANE COURSE CHANGE DETECTION ALGORITHM

In the wide field of possible outdated map content we focused our research effort on the identification of altered lane courses, as such information is highly relevant for the automated vehicles driving performance and safety. Based on our literature analysis (Section 3.2) we identified a huge and currently unused potential in the usage of additional, ubiquitously available carry-on devices, with internal low-cost sensors to detect such changes.

Mobile devices

The most prominent example for that kind of devices are smartphones and wearables, which we assume to be with the passengers inside of the vehicle. As a generalization in our opinion any device that can provide geographically referenced information should be taken into consideration to improve the performance of the time-critical detection task. The GNSS location accuracy of one single, low-cost sensor alone is not very precise and therefore as shown in the Related Work can only be compensated through the aggregation of multiple sensor readings, which is a time consuming process. As time is our major performance metric instead we propose an intelligent combination of the individual readings and the fusion of different types of low-cost sensors as presented in the following Section 5.1.2. That way sufficiently precise information can be retrieved from much less sensor readings (less time) to identify changes in the course of lanes as evaluated in Section 5.3.

Sensor fusion

By leveraging external sensor equipment available in the stated mobile devices, also older vehicles, without any own internal sensors, get enabled to participate in the data collection process. Therefore even with an assumed high market penetration rate of self-driving vehicles the amount of time required to detect erroneous map data can be improved, due to the redundancy of available sensor measurements. Furthermore drivers and passengers of normal production vehicles can also benefit from the generated change warnings, e.g. through an application on their smartphone, which increases the overall driving safety.

Following this initial idea we present our concept for a fast and robust lane-level change detection algorithm. As initial processing step in the context of the algorithm we enhance the performance of a well established GNSS trace clustering approach by fusing it with further information from additional sensors as explained in the following Sections 5.1.1ff.. The clustering results are used to identify lane center lines, which reflect the current course of the lanes. By comparing these results with past ones our algorithm identifies the changes as described in Section 5.1.4.

5.1.1 *Common Clustering Approach*

A common clustering procedure to identify lane center lines, as illustrated by Figure 23a, is used throughout several Related Works [46, 135, 144] (Sec. 3.2.2). As described in [144], the entirety of collected GNSS traces along one road direction is first segmented equidistantly. Therefore, one trace out of the whole set is randomly selected as reference line. As next step equidistant, perpendicular lines are drawn to this reference trace to intersect with all the other traces. The distance between the perpendicular lines is configurable (e.g. every 50 meters).

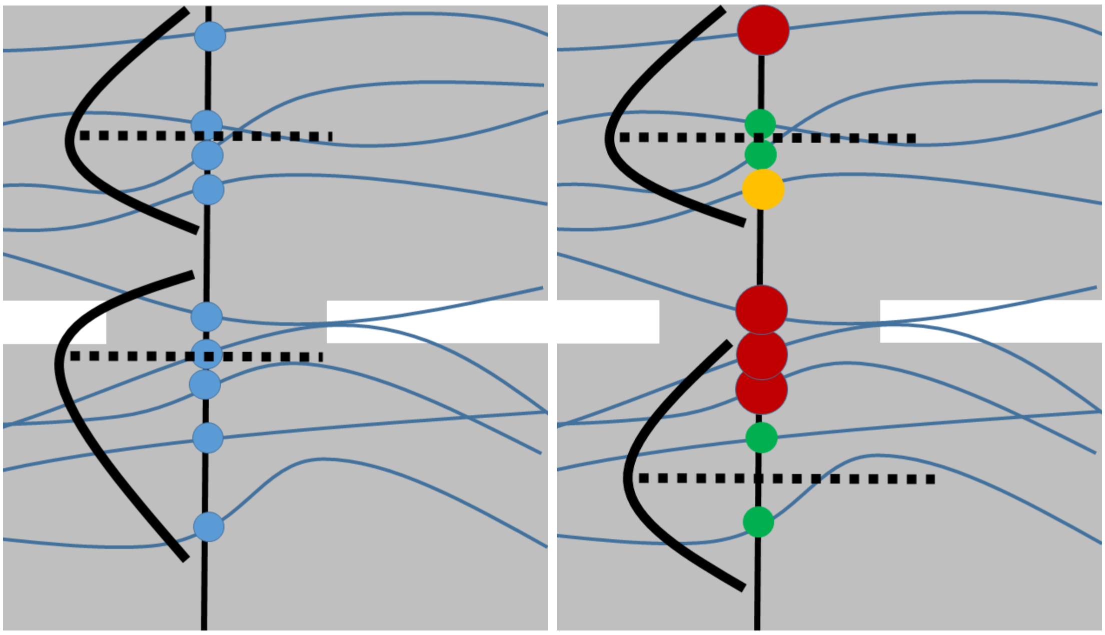
Identification of cluster centers due to point distribution

The set of intersection points (blue color) of the GNSS traces on each of these lines are then clustered. Different algorithms such as a Gaussian Mixture Model [144] or the Kernel Density Estimation Algorithm [46, 135] are available for this purpose. Through the initial intersection the degree of freedom is reduced from 2 dimensions (latitude and longitude) to only 1 dimension (along the extend of the segmenting line), which enables a faster calculation of the clustering results. The identified cluster centers (intersection with dashed line) are then considered as the lane center points and concatenated with each other to resemble the currently estimated lane curvatures.

To require less GNSS traces (and consequently less time) to achieve the same level of clustering quality as this base algorithm we annotate the GNSS points with additional sensor information as explained in the following Section 5.1.2.

5.1.2 *Proposed Weighted Clustering Approach*

As the detection of a change in the course of a lane is a time-critical task, our first goal was to reduce the high number of traces [46, 135], which are required to achieve a reliable lane-accurate clustering result. Consequentially, we did not only rely on



(a) The common clustering approach traits all location information equally (blue color). It identifies the lane centers only based on the distribution of points. The exemplary identified lane centers are marked by dashed lines (inspired by [144]).

(b) The proposed weighted clustering approach extends the common approach, by including weighting parameters regarding the accuracy of the collected GNSS point data indicated by color (red, yellow, green / low - high) and radius (the smaller the better).

Figure 23: Illustration of base line and proposed lane center calculation approaches [89].

GNSS traces / points weighted according to quality

the location information of the GNSS traces, but also on the associated quality parameters to enable a quality assessment of the different collected traces. Namely we could rely on the amount of visible GNSS satellites during the recording of the GNSS point as well as an accuracy estimation of its suggested position (Position Dilution of Precision). Both information is provided to us via the API of the Android Operating System installed on the used smartphones. Android thereby specifies the position accuracy parameter as a one sigma (68%) reliability estimation of the radial horizontal accuracy[15] (Horizontal Dilution of Precision - HDOP) (as explained in Section 2.4). The calculation of this accuracy estimation however is at the discretion of the device manufacturer and therefore can differ from device to device. To compensate a possible manufacturer specific influence, we relied in our evaluation on a set of multiple different smartphones from different brands, as explained in Section 5.2. In contrast to the previously described baseline clustering algorithm (Sec. 5.1.1) our extended approach does not select a random trace as initial reference line. Instead the trace with the highest quality out of the set of available traces is selected. We assume that this trace will resemble the overall course of the investigated road the best and thus

---

[15] https://developer.android.com/reference/android/location/Location.html (Last accessed on August 1, 2019)

optimizes the segmentation procedure by creating more accurate segmentation lines. Therefore the quality parameters of each point of the GNNS traces are taken and their average is calculated. The trace with the highest average number of visible satellites and the lowest average value of Horizontal Dilution of Precision is then selected as reference trace. This is summarized in Formula 3, were x is the reference trace and M is the set of all traces. Each trace has got its own number of N_i GNNS points.

$$\text{x is reference trace of M} \Leftrightarrow \forall y \in M : \\ \frac{\sum_{n_x=0}^{N_x} \text{no. of satellites}}{N_x} > \frac{\sum_{n_y=0}^{N_y} \text{no. of satellites}}{N_y} \cap \frac{\sum_{n_x=0}^{N_x} HDOP}{N_x} < \frac{\sum_{n_y=0}^{N_y} HDOP}{N_y} \tag{3}$$

This weighting procedure is continued in the segmentation itself. Because not all intersection points of the GNSS traces and the segmentation lines fall together with actually measured GNSS points provided by the smartphones, virtual intersection points with an artificial set of quality parameters have to be generated for those intersection points. The quality parameters of these - not actually measured - virtual points are derived from the values of their closest, actually measured preceding and succeeding GNSS point as described by Formula 4.

$$metaValue_{intersectionPoint}(acc, \text{no. of sattelites}) = \\ \frac{metaValue_{pre} * d_{pre} + metaValue_{succ} * d_{succ}}{d_{pre} + d_{succ}} \tag{4}$$

The artificial parameters are calculated based on the actual values of each of the points and weighted by their euclidean distance to the intersection point. That way our subsequent clustering algorithm can rely on this newly generated set of quality assessed intersection points.

The final cluster center, which represents the geographic position of the lane center is then calculated based on a weighted mean calculation of the different GNSS intersection points, as shown in Formula 5.

$$clusterCenter(lat, lon) = \frac{\sum (lat, lon) * f_w(accuracy, \text{no. of sattelites})}{\sum allweights} \tag{5}$$

No. of satellites and HDOP

The usage of this rather simple clustering algorithm aims at realizing shorter execution times compared to a more complex algorithmic approach such as the Kernel Density Estimation and further allows an easy introduction of the weighting parameters in the clustering procedure. We do not consider our selected quality parameters (no. of satellites and HDOP) as a finite set. Further parameters can be included into the actual weighting function of the GNSS points importance (f_w). We consider this as a promising part of possible future work. A further extension of the weighting function for example could be introduced by the consideration of the device name to identify certain used hardware components with regard to their performance.

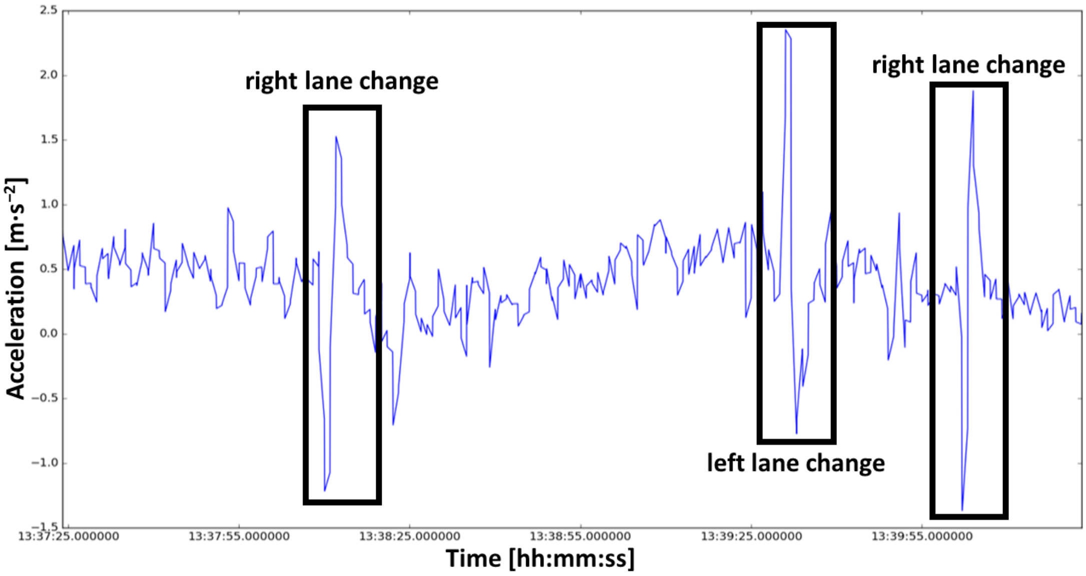


Figure 24: Behavior of the accelerometer, when performing lane changes [89].

The impact of our selected parameters in different kinds of weighting functions is evaluated in Section 5.3.3. That way we identify the most suitable weighting function to achieve the best possible performance in our personal evaluation scenario, as presented in Section 5.4.

The overall procedure is further illustrated in Figure 23b, where the quality of each of the GNSS intersection points is indicated by their color and size. A red circle with a large radius resembles a poor position estimation, whereas a green circle with a smaller radius indicates a higher quality position estimation. By relying on this quality-weighted clustering approach we achieve a lane-accurate clustering result with less traces, compared to an approach, that only relies on the location information of the GNSS data, as verified in our evaluation in Section 5.3.

5.1.3 *Identification of Lanes*

The exact identification of the currently available lane centers from noisy GNSS traces still remains a difficult task, even with our additional quality parameters in use.

In the related clustering approaches the individual location error of each GNSS trace has to be compensated through averaging out a large set of measured traces [135].

For our proposed algorithm we choose a different way to significantly reduce the amount of required traces to identify the number of lanes for which the GNSS data has to be clustered. We rely on an algorithm, which identifies the lane changes of the vehicles during their trip, similar to the ones used in presented Related Work (Sec. 3.2.3). This initial lane identification enables a pre-mapping of the set of available GNSS points upon the set of available lanes. In the following clustering step only the lane-individual sets then have to be considered for the overall end result.

Lane ID derived from consecutive lane changes

As shown e.g. by the works of Aly et al. [47] and Wu et al. [49] it is possible to derive the current lane ID of a vehicle only by using low-cost sensors, which are available in a smartphone, namely the accelerometer and the gyroscope. Without knowing the initial lane of the vehicle, the probability distribution (over all available lanes) of its current lane position is narrowed down quickly over time through the detection of consecutive lane changes. In the end only one possible lane remains.

The example Figure 24 shows the obtained sensor readings from the accelerometer for several consecutive lane changes. The sampled sensor data is shown without any noise filtering, showcasing the capabilities of this sensor. The lane change detection itself is implemented rather simple, based on a static offset gate of the measured min and max values per time interval.

In Section 5.3.2 we compare the performance of our algorithm-based lane identification with a manually annotated ground truth collected through our custom android application, as described in Section 5.2 from the driver's input.

We achieve similar clustering performance results with both data sets (manually annotated ground truth and algorithmic annotated), even though our algorithm's lane change detection performance could be further enhanced by the suggested additional techniques of Section 3.2.3 (e.g. bootstrap and organic anchors such as traffic rules and potholes).

Through the highly reliable performance results shown in the Related Work and gained in our own evaluation, we consider the work of [47, 49] as a fundamental contribution for our personal lane course change detection algorithm and an important quality feature to reduce the amount of required traces for lane-accurate clustering results. Through the identification of the correlating lane ID of each GNSS trace's segment, we are able to prefilter the data accordingly to its expected lane. In consequence, the influence of the Gaussian Noise experienced by the GNSS receiver can be heavily reduced in the clustering process, as shown in Section 5.3.1. This provides our approach with a significant advantage over related clustering concepts, such as the Kernel Density Estimation Algorithm, which rely on nothing else, but the GNSS location information.

5.1.4 *Detection Step for Lane Course Changes*

Based upon the aforementioned enhancements of the clustering procedure the second processing phase of our algorithm takes place: the detection. The detection of lane course changes is realized as illustrated in Figure 25 by comparing the most recent lane clustering result with the set of historic results. At the beginning each newly incoming GNSS trace, annotated and pre-filtered for its specific lane as described in the Sections 5.1.2 and 5.1.3, is added to the set of available GNSS traces. Initially, this collection is empty. To ensure robust clustering results and in consequence a reliable change detection, a minimum of X traces has to be collected initially, before any further processing step can be executed. That way our detection algorithm reduces the influence of negative impact factors, such as faulty sensors or Gaussian noise, down to a reasonable level. Based on our working experience in the Ko-HAF project (see

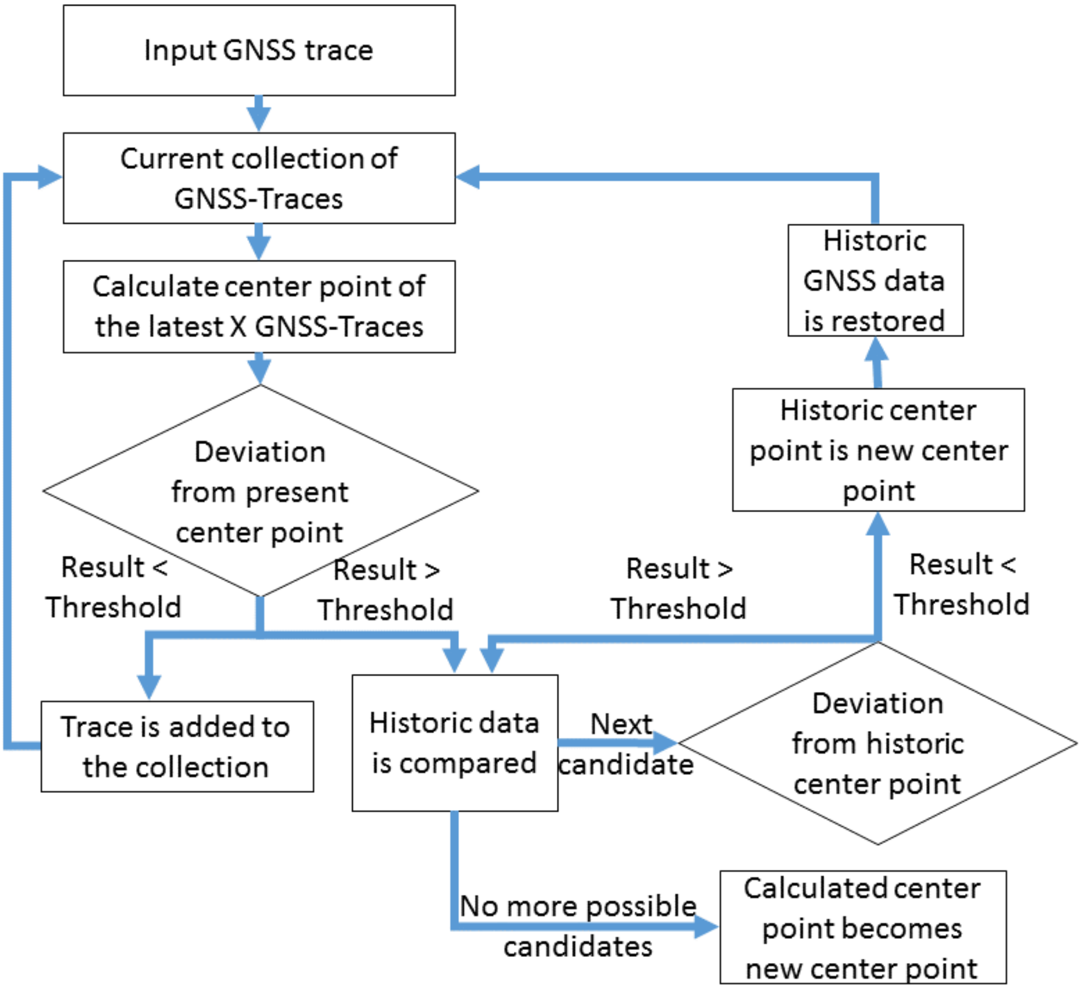

Input GNSS trace
Current collection of GNSS-Traces
Calculate center point of the latest X GNSS-Traces
Deviation from present center point
Result < Threshold
Result > Threshold
Trace is added to the collection
Historic data is compared
Next candidate
Deviation from historic center point
Result > Threshold
Result < Threshold
Historic center point is new center point
Historic GNSS data is restored
No more possible candidates
Calculated center point becomes new center point

Figure 25: Deviation detection process [89].

Section A.2), we consider a clustering accuracy on lane-level as minimum requirement to ensure the safety (not the level of comfort) of self-driving vehicles, which rely on an HD Map. The exact number of required traces X is scenario-dependent and has to be identified individually based on the accuracy requirements on the clustering results, the overall quality of the used sensor equipment and the amount of incoming traces over time.

For our personal data set a reasonable value for X is investigated in Section 5.3.5. For each trace, which surpasses this limit X, the trace itself and the last X traces are considered as input for a new cycle of our clustering process. The influence of the last X traces thereby is weighted according to the time, when they were collected and their overall quality, as shown in Formula 6.

$$\text{importance of trace t} = w_{meta} * e^{\left(\frac{(time_{trace\_t} - time_{newest\_trace})^2}{degradation\_factor}\right)} \quad (6)$$

Each trace's importance degrades over time, with the oldest traces being considered the least valuable ones. The $degradation\_factor$ is configurable depending on the frequency of newly incoming traces and required clustering accuracy. In our evaluation we selected a $degradation\_factor$ through which one week old traces were only treated with half of their initial importance. Besides the time factor, also the overall quality of each of the traces is considered through a weighting function w_{meta}. Again, the influence of the different quality parameters in the weighting function should be adapted individually on the used set of parameters and their overall impact on the clustering algorithm's performance. For our own set of quality parameters we investigate various weighting functions in Section 5.3.3. That way we are able to find one with high overall performance, considering our test scenario's data set of GNSS traces.

Continuous reevaluation

The clustering procedure is then executed by recalculating the lane-specific center point of each lane segment along the track using this set of $X + 1$ traces. For each individual lane, the consecutive sequence of its specific center points on each segment represents the currently assumed curvature of the lane itself. On a two lane highway, for example, two separate lane center points per segment have to be calculated. The detection of lane course changes therefore can be visualized with an accuracy based on the distance between the segments (see Figures 34 and 35).

As the next step to detect the changes along the route, the current set of center points, which is still considered correct is compared with the newly generated center points set. If both sets deviate from each other in one or more points by more than a certain threshold T, the affected segments are identified to have changed. In this case, the old set of lane center points is erroneous and gets replaced by the newly calculated one. Afterwards, the old set of center points is stored away in a collection of historic lane center point sets. If this collection is not empty, there exist other previous clustering results, which might resemble the currently changed road course. In consequence, the newly calculated set of center points is also compared with all of these sets. If one of these sets matches with the newly calculated one, it is considered as the restored road course. Such a situation can occur for example, when a construction site has changed the overall road course, but was completed in the meantime. Then all the changes are withdrawn to the road course, previously to the construction site, and the old trace data becomes reusable. That way we enable our algorithm to not have to recollect the amount of traces T over and over again. Instead it can generate more reliable results quicker, as the old trace data can be reused if applicable. We consider this feature especially useful if vehicles do not drive frequently in such areas and consequently only a small amount of traces can be collected over time, compared to a dense traffic environment.

The exact selection of the threshold value T as the minimum accepted deviation distance between two consecutive clustering cycles is subject to individual choices regarding the required performance of the algorithm. For our personal evaluation we assumed a value of $2/3 *$ normal lane width for T. We considered only $2/3$ of the lane width as reasonable threshold value, as the distance between the identified lane center lines can be diminished under the influence of certain events on the road, e.g. a construction site. The normal lane width is either regulated by federal authorities[16] as it is in our evaluated scenario (Sec. 5.3), or can be correlated with the average width of a vehicle. Again this correlation should be made specifically for certain countries or traffic scenarios (highways, city streets) as the common width of a lane can differ from country to country (e.g. comparing German highways vs. highways in the USA).

Average speed as additional criteria

Besides the longitudinal deviation along the course of the road also the latitudinal extent of the detected road hazard has to be visualized. Only through a timely warning, ahead of the potentially dangerous road section, the safe operation of self-driving vehicles can be maintained. Our algorithm realizes this goal by additionally calculating the average traveling speed derived from the collected GNSS traces for each of the segments. As the traveling speed is regulated before and after a construction site, e.g. from a maximum speed of 120 km/h down to only 80 or 60 km/h within the construction site and back again to the previous value afterwards, it provides a good indicator for the latitudinal extent of the road hazard. Our lane course change detection algorithm only indicates changes if, besides a latitudinal deviation of the average speed, also a longitudinal deviation could be detected. A deviation of speed for example might also be caused by a traffic jam in the morning rush hour and therefore alone is not a sufficient criteria. The change detection algorithm's final visualization result of the detected construction work is exemplary shown in Figure 34 for our evaluation scenario .

5.2 CREATION OF TEST DATA SET

At the beginning of our work no suitable test data set for the evaluation of the change detection algorithm was publicly available. To resolve this problem we conducted our own measuring campaign, which spanned from April 2016 to January 2017, to collect the required test data. Therefore several hundreds of test drives have been performed.

Smartphones as low-cost test devices

To mimic an actual large-scale deployment on low-cost after market devices, we used several different smartphone models of different brands as measuring devices throughout the campaign. Namely 30 different devices of 12 different types including e.g. Nexus 4, Nexus 5, Blackberry Classic and Samsung Galaxy S7 were used. All these different models contain sensory equipment of different quality levels. This resembles an actual future deployment phase where also various devices will provide their data. During the measuring test drives the smartphones have been placed on the dashboard of the test vehicles. No further sensor calibration was performed to resemble a realistic application scenario of the end-customer. Through the usage of our custom built Android application we were able to collect a data set of over 1.934.000 GNSS points.

---

[16] https://www.forschungsinformationssystem.de/servlet/is/275112/ (Last accessed on August 1, 2019)

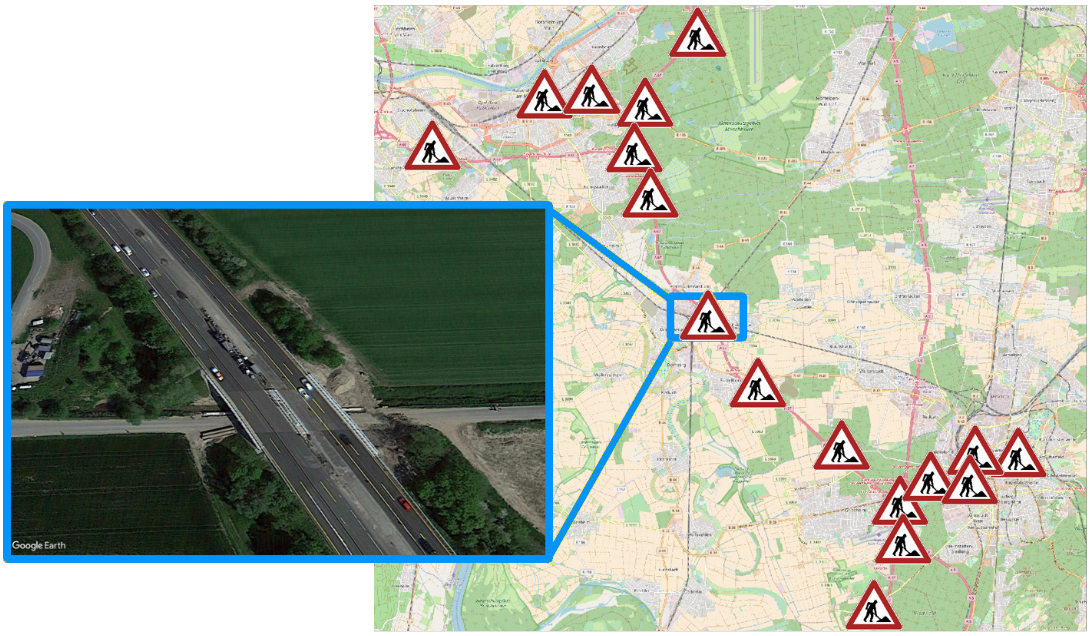


Figure 26: Visualization of construction sites contained in the data set collected in test drives from April 2016 to January 2017. The evaluated example construction site scenario is marked in the map. The related satellite images were taken on the 5th May 2016. Satellite image ©Google Earth, map data ©OpenStreetMap contributors

date	hour	accuracy	longitude	latitude	modelname	device imei	annotation	vehicle speed [m/s]	number of satellites	lane ID
04.05.2016	10:10:05:0184	11.0	8.43290705	50.00252679	Nexus 5	3521...1314.0	start of construction site	30.75	8	1

Table 6: Example of an annotated GNSS point out of our collected data set for the evaluation of the lane course change detection algorithm.

Each GNSS point was annotated with our considered meta-information. Namely this is the current lane of the vehicle, which is derived from the detected lane changes of the vehicle based on accelerometer and gyroscope readings, the number of available satellites, when the GNSS point has been estimated and the related GNSS position's accuracy value (HDOP). Furthermore we manually annotated the start and the end of experienced construction sites with the push of a button throughout the different test drives as reference for our evaluation. An excerpt of the data set is shown in Table 6.

Own dataset published

That way we were able to collect GNSS trace data for over a dozen different construction sites, located in urban, suburban and rural areas as shown in Figure 26. This data was then stored in a database for the further processing steps. We made this data set publicly available on Github[17]. It provides reference and benchmark material for future scientific work and related algorithms in this research field. The benchmark

---

[17] https://github.com/florianjomrich/construction_side_traces_fjom (Last accessed on August 1, 2019)

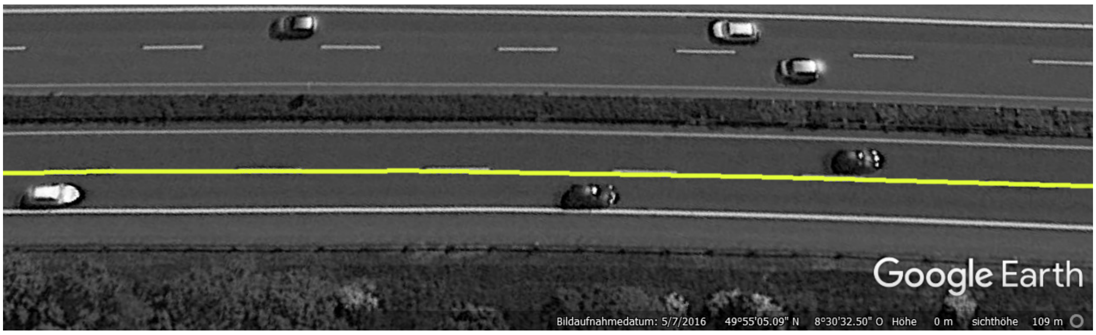

Google Earth
Bildaufnahmedatum: 5/7/2016 49°55'05.09" N 8°30'32.50" O Höhe 0 m sichthöhe 109 m

Figure 27: Reference line created from satellite images of the center dashed line [89]. Satellite image ©Google Earth

material contains, besides the device model name with which the specific GNSS data point has been collected, further detailed information regarding the weather and the temperature at the day, when the test-drive has been conducted. We did not include those values as input into our presented evaluation process, but consider it as a highly relevant source of data for future improvements of the achieved results. For example the device specific filtering of the input data could improve the overall achievable quality furthermore, this however possibly requires additional measuring campaigns to retrieve an even more diverse set of data points.

The evaluation of the achieved clustering performance and the lane course detection as expressed in the following Section 5.3 are based on a construction site on the German highway A67 between the cities of Rüsselsheim and Darmstadt, which is marked blue in Figure 26. GNSS traces in both directions of the two lane highway have been collected - 369 traces for the direction from Rüsselsheim to Darmstadt and 292 in the opposite direction from Darmstadt to Rüsselsheim.

By using this bigger data set we ensured to have a sufficient high amount of GNSS trace data to showcase the benefits of our approach in comparison to the Related Work regarding the amount of required traces.

5.3 EVALUATION OF CLUSTERING PERFORMANCE

The performance of our proposed clustering process, described in Section 5.1.2ff., is the fundamental base for the successful operation of our subsequent change detection step (Sec. 5.4).

As previously stated we consider lane-accurate clustering results as the minimum degree of accuracy that a clustering result has to achieve for our change detection to work properly and to ensure the safe operation of highly automated vehicles.

KDE as reference algorithm

Based on our literature analysis we selected the Kernel Density Estimation algorithm (KDE) as our reference algorithm for the performance evaluation of our proposed weight-based clustering concept. The KDE was selected as it is the most frequently used common clustering approach in the investigated Related Work. It achieves results well above the level of lane accuracy, if sufficient amounts of traces are provided. To compare the performance between the KDE algorithm and our own weighting algorithm, we relied upon a reference center line, which we draw based on satellite images (see Fig. 27) of the investigated two lane-highway A67 (see Sec. 5.2). This evaluation setup is motivated and inspired by the recent work of Pauls et al.[131], which state the availability of high-resolution aerial images as a reliable and easily accessible source for geo-referenced data, specifically for the verification of HD Map data. The reference line overlies the course of the center dashed line, which separates both highway lanes from each other as illustrated by Figure 27. Through this design choice the deviation between the algorithmic estimated lane center lines and this reference line became our performance metric for the achieved clustering results described in the following. This initial evaluation was conducted on a section of the two-lane highway A67, where no construction sites were present throughout the time of our measuring campaign. That way we could rely on federal specifications to accurately evaluate the clustering results. The common width of one lane on a German two-lane highway[18] (see Sec. 5.1.4) is 3.75 meter.

In consequence of the lane width an optimal clustering algorithm thus would achieve a longitudinal distance of 1.85 meters between all its calculated lane center points and our reference line in the middle between the two lanes. This optimum is indicated by the two dashed lines in Figure 28ff., one for each lane.

The x-axis of our plotted evaluation results (e.g. Figure 28) indicates the deviation of the lane-clustering results from our reference line (thick center line). Thereby the aggregates of all segment results are represented through box plots. A deviation from the reference line to the left/right, for the algorithmic estimation of the left/right, dashed lane center line, is indicated by positive/negative values in meter.

In a first evaluation step described in the following Section 5.3.1, we investigated the impact of the lane ID, as only available meta information, on the overall achievable clustering performance.

5.3.1 *Impact of the Lane Annotation*

The common clustering algorithms presented in the Related Work (Sec. 3.2.2) only rely on a large amount of GNSS location information to achieve their clustering performance. Our personal approach instead allows the lane-specific filtering of the GNSS data before its further processing by the subsequent clustering algorithm.

In this subsection we investigate the performance impact of this design decision on the required amount of traces, the correlated amount of time required for their collection and the overall achieved performance of the final clustering results. Improved clustering results provide the basis to increase the reliability of our personal change detection algorithm.

For this initial performance comparison we relied on a simple mean calculation as our clustering algorithm, not using any of the additional weighting parameters and functions as investigated in Section 5.3.3. That way we were able to evaluate the influence of the lane-specific pre-filtering on the final clustering results.

---

[18] https://www.forschungsinformationssystem.de/servlet/is/275112/ (Last accessed on August 1, 2019)

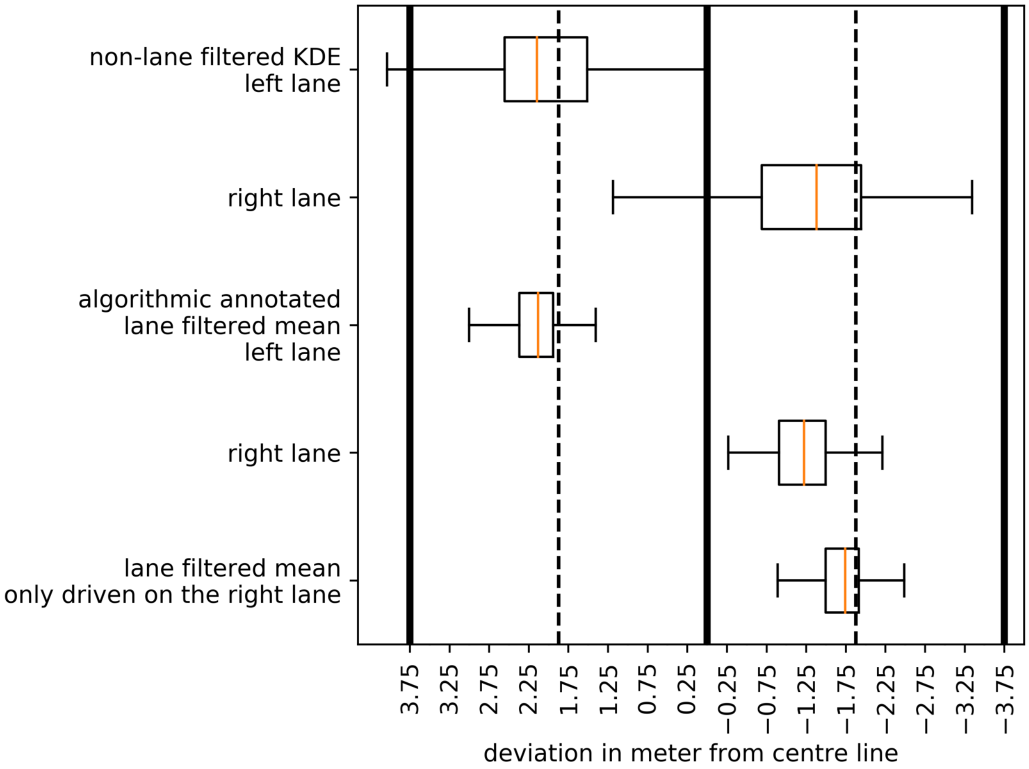


Figure 28: Juxtaposition to identify the impact of lane-specific filtering of GNSS data on the clustering results. Our compared algorithms are the chosen reference, the Kernel Density Estimation algorithm (using the non-lane-filtered data set as input) and a simple, non-weighted mean calculation of the center point (using the lane-filtered data as input) [89].

In comparison we executed the Kernel Density Estimation algorithm on the non-lane filtered data set and compared the achieved performances of both approaches with each other as shown in Figure 28.

To achieve a clean separation of the different sub-segments of the GNSS traces onto their related lane IDs, our clustering approach performs an initial cleaning step of the GNSS data. Thereby all triplet sets of recorded GNSS location points, which have been collected during, right before or right after a lane change of the vehicle, get removed from the data set. That way our approach ensures to remove their potential negative impact on the overall clustering performance.

Positive impact of lane-filtering

For both lanes the achieved results clearly visualize the significant impact of our initial lane-filtering step on the overall clustering performance. The variance of the cluster centers identified by the Kernel Density Estimation Algorithm, as indicated by the boxplots in Figure 28, is much higher than the one of the simple mean calculation, which only had to identify a single center point for each of the given lane specific data sets. Some instances of the Kernel Density Estimations cluster centers even lie outside of our given lane boundaries, as specified by the federal regulations authorities.

The results of our own lane-filtered mean in contrast achieved a much better overall variance and stayed within the lane boundaries for all 225 investigated segments along our considered track on the highway.

Against our expectations the obtained results for the identification of the right lane center lines of both algorithms possessed a visible static offset towards the road center. The average of the identified cluster center points for the left lane in contrast were close to the actual center lane as expected by us.

To further investigate this phenomenon, we conducted 37 additional test drives, for which the drivers were told to stay on the right lane throughout the whole time of the trip. Afterwards, we used this newly generated data set as input to our clustering algorithm. The obtained results therefore are shown in the lowermost plot in Figure 28. It is clearly visible that these results are very similar in terms of variance around the lane center lines, compared to the results obtained for the left lane. In consequence we suppose that the difference in the obtained clustering results for the right lane can possibly be tracked back to a difference in the overall driving behavior throughout the measuring campaign. As our investigated highway scenario posses two lanes, this is probably due to the different average driving speeds, that can be achieved on the individual lanes. The test drives have been executed via the usage of sedans of the Opel fleet. The right lane however was mostly occupied by slower driving trucks. In consequence the drivers passed these slower traffic participants on their travel, due to their driving habit. Consequently our measuring vehicles stayed for a much longer time straight in the left lane or quickly changed back and forth from the right lane to overtake the slower traveling trucks. These quick overtaking changes could have an effect on the obtained GNSS trace data, as the Android smartphone itself is using a Kalman filter to smooth the obtained GNSS location points. In our opinion the Kalman filter probably smooths out these quick location changes due to the overtaking maneuvers. This in consequence then could lead to a shift of the location data for the right lane data, which is separated from the rest of the trace due to the lane changes. This identified aspect motivates future work in this area to further improve the achievable trace quality and overall clustering performance. The speed of the vehicles for example could be a good indicator to reduce the influences of lane changes on the trace data. Slower moving trucks for example probably will stay in the right lane and therefore should be prioritized for the clustering process of this lane.

In the following we present our evaluation results based on the data gathered from the left lane of the highway. The similar results achieved for the right lane are shown in Section A.7 of the Appendix.

5.3.2 *Algorithmic vs. Manual Annotation*

Besides the overall impact of the lane annotation on the clustering results, we also wanted to compare the performance of our rather simple lane detection algorithm with a ground truth data set. This comparison was conducted to verify the performance and thus the possibility of a future large scale deployment of our proposed approach. The ground truth data set thereby was generated through the manual push of the

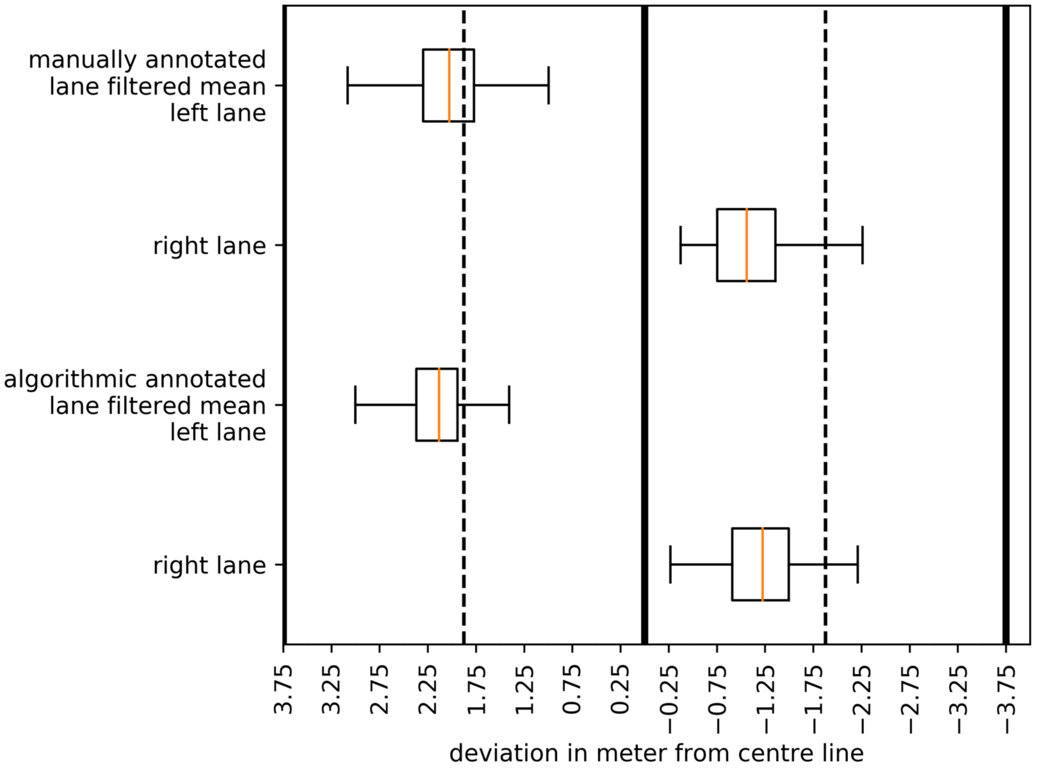


Figure 29: Performance comparison of our simple mean clustering algorithm relying on the algorithmic annotated lane IDs in comparison to a manual annotated ground truth. The shown results for both lanes are again obtained from all 369 available traces collected on the highway A67 in the driving direction between Rüsselsheim and Darmstadt [89].

according buttons in our custom Android application, as explained in Section 5.2, each time a lane change was executed. The obtained evaluation graph therefore is shown in Figure 29. From the four plots it is clearly visible that both approaches performed similarly well. There is no clear performance difference visible, as the manual annotated data set performed slightly better on the left lane, with respect to the average calculated lane center point (orange bar). On the right lane instead the algorithmic annotated data set achieved a slightly better performance. We are very pleased with these obtained results, as our own lane identification algorithm was conducted via a rather simple implementation using static thresholds. Thus, we are confident that these performance results can be further improved in future work. The threshold values of the accelerometer to detect a lane change for example could be adapted accordingly to the current speed of the vehicles to improve the performance. Furthermore the proposed improvements of the Related Work (Sec. 3.2.3), such as boostrap and organic anchors (e.g. traffic rules and pot holes on the street) should be considered too.

Clustering results of algorithmic and manual annotation similar

5.3.3 *Comparison of Different Weighting Functions*

Motivated by these positive evaluation results, we investigated the performance of several different weighting functions (Formula 5), to boost the quality of the final lane clustering results even further. Our two further considered quality parameters: the number of visible satellites and the estimation value of the horizontal dilution of precision, thereby were considered either as single input values or as combined sum for the different weighting function, as shown in Figure 30. The number of satellites thereby was considered as proportional input into the weighting function, as more satellites tend to achieve a better localization result. The estimation value of the Horizontal Dilution of Precision instead was considered anti-proportional in the weighting, as a smaller accuracy value resembles a better location estimate.

For the comparison of the different functions, we again considered our full data set of GNSS traces as used in the previous Sections 5.3.1 and 5.3.2.

The obtained results, illustrated for the left lane in Figure 30, show that both quality parameters improve the clustering quality best in terms of achieved variance and average, if considered with high importance.

Quality parameters improve clustering result

Out of the complete set of investigated weighting functions, $\mathtt{NumberofSatellites}^3$ and $2^{\mathtt{Accuracy}}$ performed best upon our personal data set in terms of the achieved average distance between the identified lane center point and the dashed reference line. As $2^{\mathtt{Accuracy}}$ achieved an overall better variance, we selected it as our weighting function of choice for the further evaluation, especially to optimize the possible performance for our lane course change detection as evaluated in Section 5.4.

However we consider this investigation only as a first glance into the general consideration of a suitable weighting function. Further optimization potential is considered for future work, e.g. due to the selection and weighting of additional quality parameters, such as the type of vehicle (sedan, truck, ...) or the used measuring device (cheap or expensive available hardware equipment).

5.3.4 *Effect of Weighting GNSS Points on Lane Filtered Data Set*

In our next evaluation step we verify that our proposed meta information and our weighting procedure further improve the clustering results beyond the point that was achieved through the initial lane filtering step. To showcase their positive impact, we compared the overall performance of our three distinct algorithms (mean, KDE and the weighted mean) with each other. Each algorithm was executed on the full lane-filtered data set.

The obtained clustering results for the left lane are shown in Figure 31. As expected, the Kernel Density Estimation algorithm also performed better on the lane-filtered data set, compared to its previous execution on the non-lane-filtered data set. The Kernel Density Estimation algorithm however thereby performed similar as the much simpler mean calculation, which achieved a much better execution time in comparison.

To our satisfaction the weighted mean achieved the best performance results of all three algorithms executed on our GNSS trace data set. It outperformed the KDE and

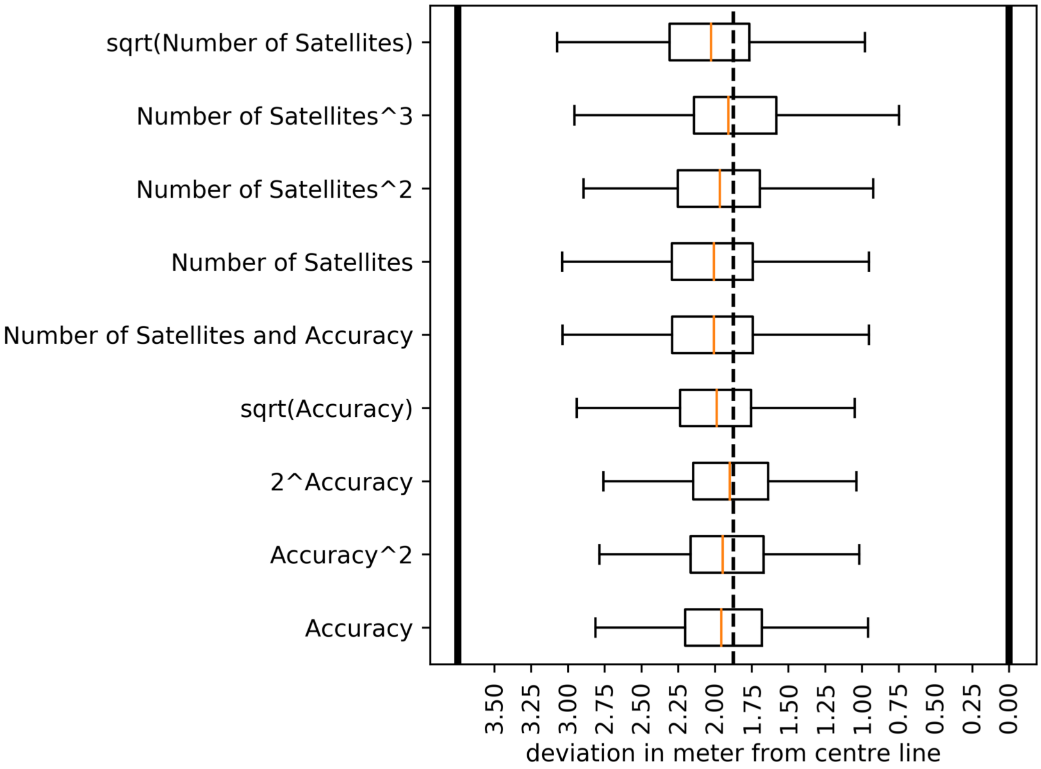


Figure 30: Comparison of different weighting functions (left lane data) [89].

the simple non-weighted mean in terms of the variance of the identified lane center points and their average distance to the lane center reference line. We are well aware that other Related Works might have achieved better overall clustering performance results as they are highly dependent on the available GNSS receiving equipment. However, through our extensive measuring campaign, we were able to verify the positive impact of the proposed weighting process, even when relying on a large set of different smartphones, which are equipped with various GNSS sensors of different quality levels.

Furthermore our approach did not to rely on any additional side conditions regarding the lane center points, e.g. as done by Neuhold et al. [46], which might not be fulfilled in the considered scenarios, such as construction sites and accidents.

5.3.5 *Performance of Weighted Clustering for Different Numbers of Traces*

As stated previously time is the most crucial performance metric for our lane course change detection algorithm. This stands in contrast to the investigated Related Work, which focused mostly on a high overall clustering accuracy. To close this research gap we evaluated the performance of our previously selected weighted clustering algorithm under the consideration of different amounts of GNSS input traces. The impact

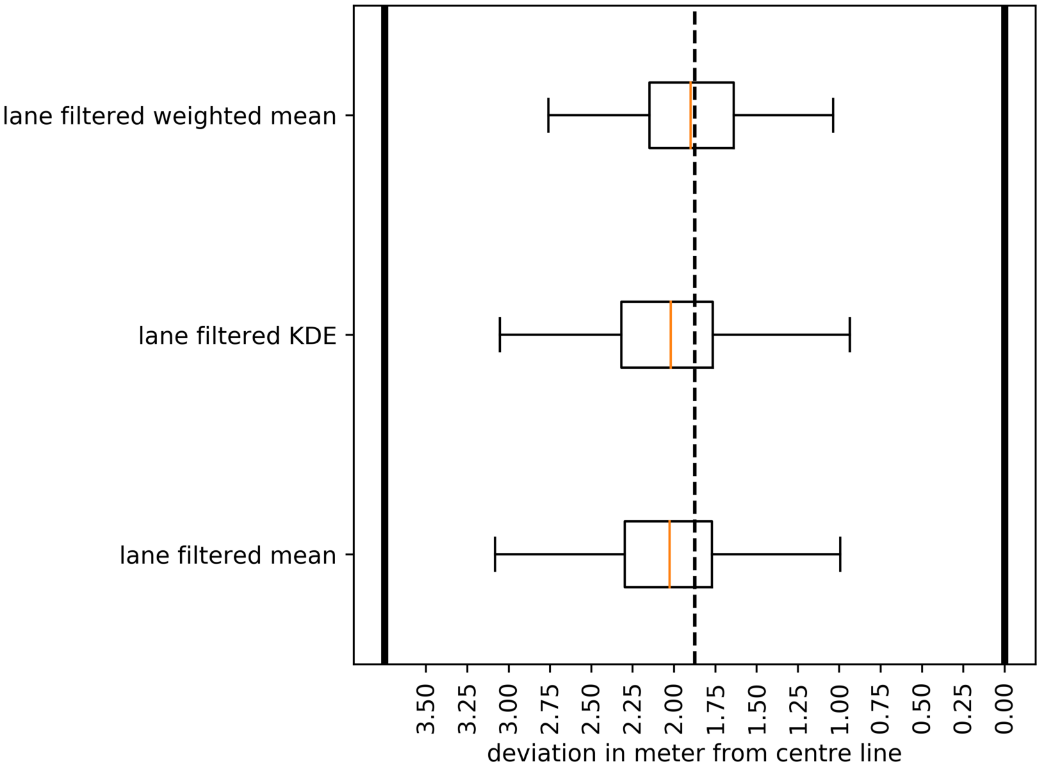


Figure 31: Comparison between weighting and non-weighting algorithms using all available pre-lane filtered traces (left lane data) [89].

of a steady increase of available GNSS traces upon the overall achieved clustering accuracy is visualized by Figure 32 for the clustering results obtained on the left lane.

For the graphs we randomly and increasingly selected a small subset between 5 and 90 traces out of the full evaluation set of 369 available traces, as input data.

Due to our working experience in the Ko-HAF project, we consider a clustering accuracy within the lane boundaries of the track as the minimum level of accuracy to let our change detection algorithm ensure the driving safety.

The different boxplots of Figure 32 indicate that this level of accuracy was reached by our weighted clustering algorithm reasonably fast at 25 randomly selected traces. By increasing the amount of available traces even further up to 90 only a slight performance improvement was achieved.

Through these achieved results we could verify the feasibility of our proposed algorithm to enable a fast and reliable lane course change detection.

The overall achieved performance can be improved even further by a pre-selection of only high quality traces out of the set of available data, as shown in the following Section 5.3.6.

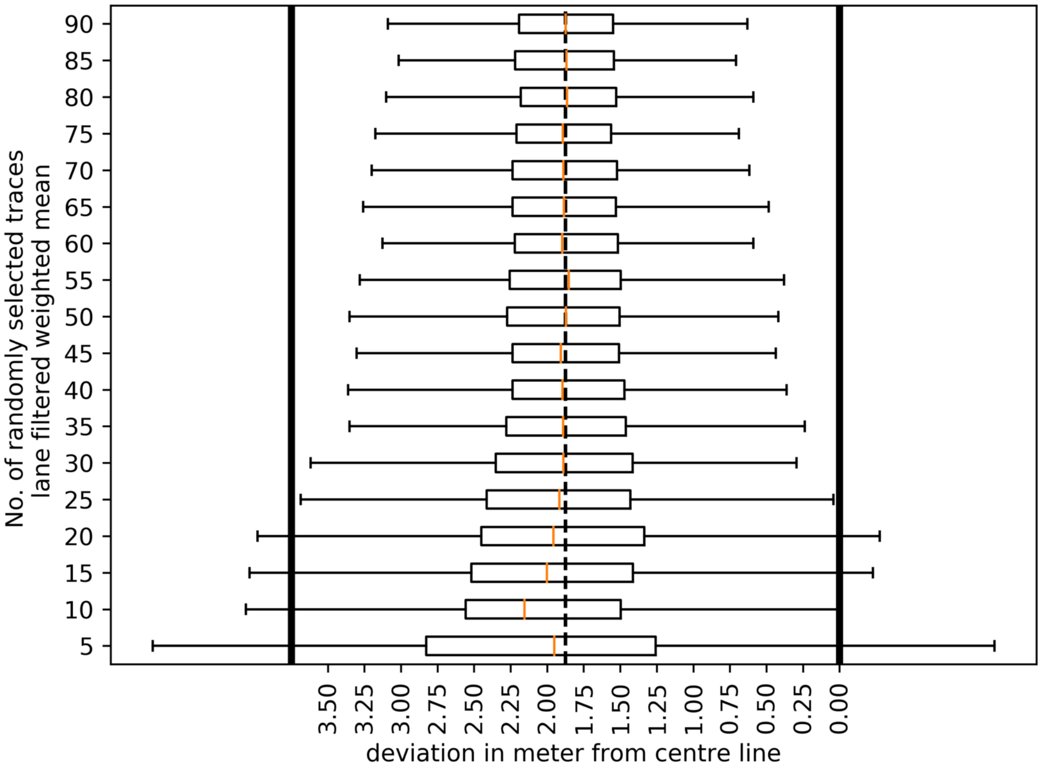


Figure 32: Influence of different amounts of available input traces on clustering performance (left lane data) [89].

5.3.6 *Effect of Trace Selection*

Motivated through the previously obtained results we further investigated the impact of the overall quality of the available trace data upon the clustering results. For this evaluation we conducted a test where we compared a subset of 70 randomly selected GNSS traces with the 70 best traces out of our complete evaluation GNSS trace set (see Figure 33). The number of 70 traces was selected, because the obtained clustering results indicated a saturation in the performance above this amount of traces. Consequently, we assumed a much higher amount of traces to be required to significantly improve the clustering accuracy even further. Thus as time is a highly important parameter for our change detection algorithm we considered 70 traces as a reasonable trade off between the required collection time for the traces and the overall achievable clustering performance. However these assumptions should be reconsidered individually for a different GNSS data set, as the threshold values might differ for a different scenario with different available measuring equipment. The subset of the 70 best traces was selected based on the average accuracy value of their intersection points with all 225 created segment lines along the considered scenarios route. The trace with the lowest sum of accuracy values is considered as the best trace of our data set. As expected, the obtained results using the 70 best traces as shown in Figure 33 outperformed the set

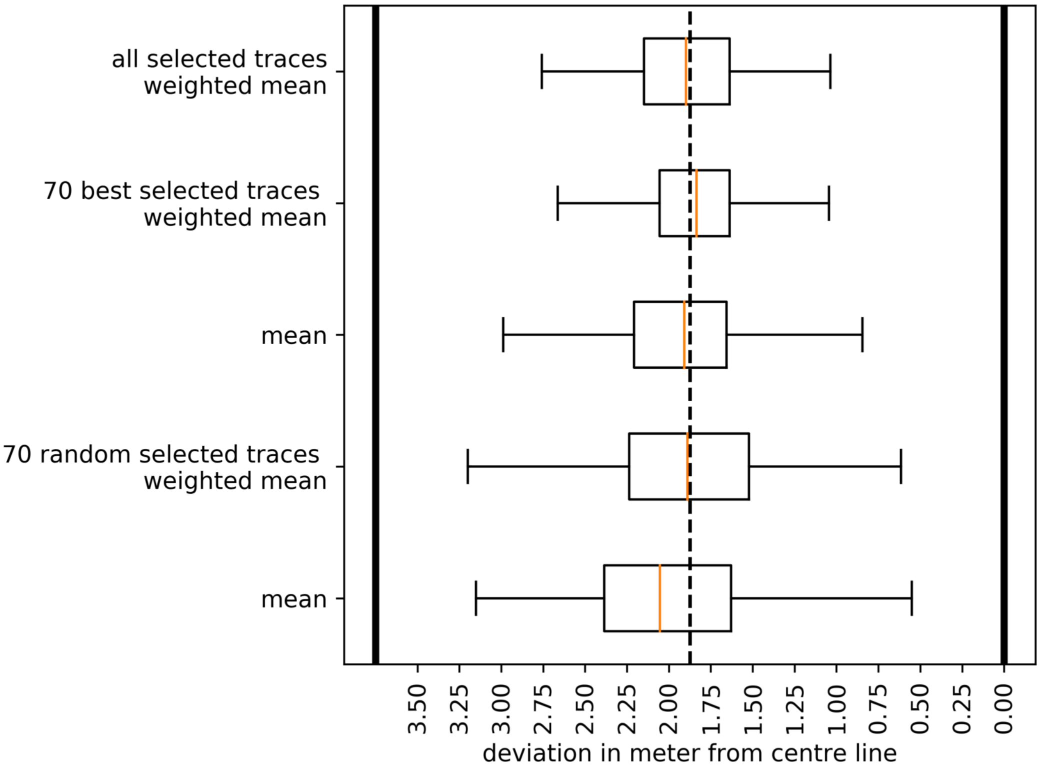


Figure 33: Performance comparison between randomly selected traces and the most accurate available traces (left lane data) [89].

Positive impact of qualitative trace selection

of 70 random traces significantly in terms of clustering performance. To our surprise the 70 best traces even achieved a similar clustering performance to the reference case, where we considered all 369 traces as input for the clustering process. This further supports our initial assumption, that the quality of the GNSS traces plays a major role in the achievable quality of the clustering results. Consequently, it might be worth to consider only a high quality subset of GNSS traces for the clustering procedure in future. If new incoming traces do not significantly differ from the currently present clustering results and are also not better in terms of average quality compared to the existing set of traces, they might as well be neglected in the future process. This assumption however requires additional investigation in future work with a larger set of traces to verify our initial observation.

5.4 EVALUATION OF CHANGE DETECTION - HIGHWAY CONSTRUCTION SITE SCENARIO

After the achieved promising performance results of our weighted clustering algorithm, we continued our evaluation by the investigation of the lane course detection performance of our algorithm.

Therefore we selected a different section of the highway A67, where a construction site has been present during the May of 2016, as illustrated by the Figures 34 and 35,

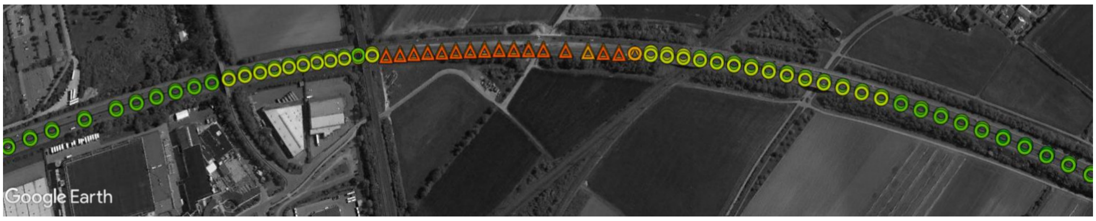


Figure 34: Correlation between the average vehicles speed (indicated by color) and the location of the construction site (indicated by triangles). Speeds right before and in the construction site range from 90 km/h down to 57 km/h (yellow - red). Before and after the construction site normal driving speeds of around 125 km/h are reached in average (green) [89]. Satellite image ©Google Earth

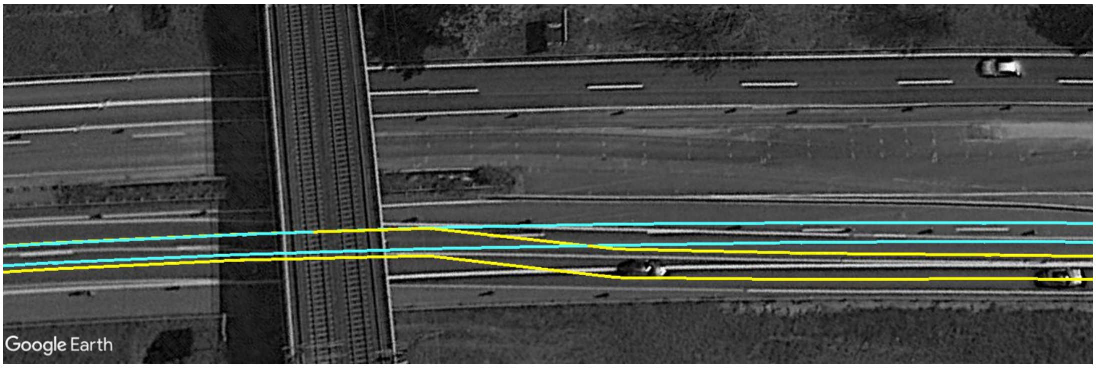
which are underlaid with satellite images provided by Google Earth. As explained in Section 5.1.4, the deviation detection algorithm identifies a change in the course of lanes in lateral and longitudinal direction. Figure 35 shows the lateral offset of the assumed road course during the ongoing construction works (yellow) and after the construction site was completed (cyan). The achieved clustering results show the great performance capabilities of our deviation detection algorithm to accurately resemble the current road course, even though the investigated scenario could only provide it with a rather small amount of GNSS traces between 15 and 25 for each lane, during the presence of the construction site event. The detection algorithm resembles the entrance and the course of the construction site well by staying in the given lane boundaries in each intersecting segment. Only at the exit of the construction site the assumed distance between the two adjacent lanes, as indicated by the two yellow lines, becomes unfeasibly small for a normal vehicle's width. To avoid such situations, we propose to rely only on a larger subset of traces, as investigated in Section 5.3.5 for our personal data set. Furthermore to reduce the distance between two adjacent clustering segments could improve the identification accuracy of the correct lane course in future work.

Construction site accurately detected

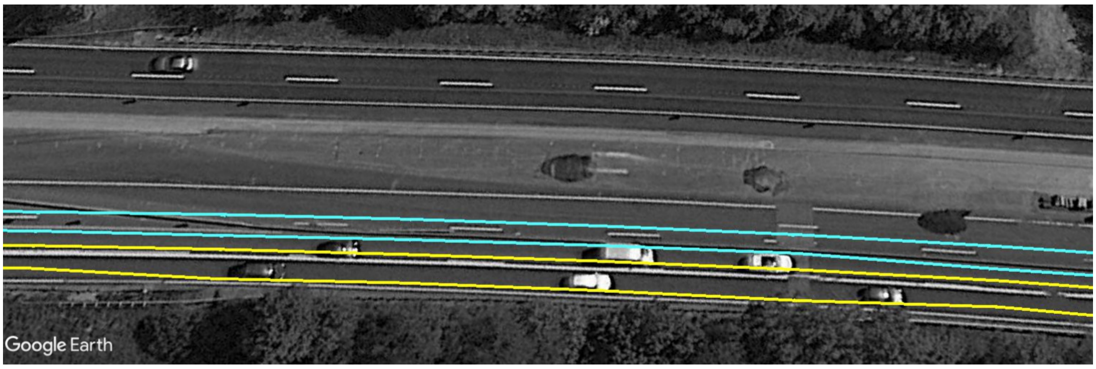
Figure 34 completes the safety relevant detection of the construction site by providing the assumed longitudinal extension of the construction site. Therefore the algorithm correlates the average achieved driving speed for all street segments along the course with the previously identified lateral offsets in the clustered GNSS traces. The identified location of the construction site is indicated by triangles on the map. The correlating achieved average traveling speed is indicated by color. The color range starts from green with average achieved speeds of 125 km/h and above over 90 km/h down to the construction site area with as low as 57 km/h in average (yellow to red color).

Overall, our evaluation results show the successful application of our proposed lane course change detection algorithm in an actual highway scenario. To identify changes it relies on a weighted clustering step, where it incorporates additional quality information from the GNSS and further low-cost sensors. That way it requires less GNSS traces as related clustering algorithms to achieve lane-accurate results. Through our extensive measuring campaign, where we relied exclusively on smartphones as measuring devices, we verified the potential of ubiquitously available after-market devices to improve the maintenance of HD Maps.

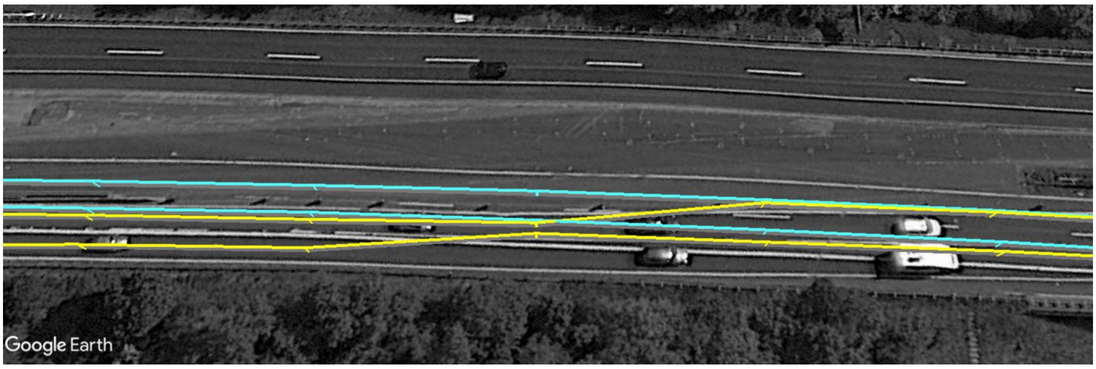
To realize a reliable data stream of vehicular sensor data and received map updates for a self-driving vehicle requires comprehensive knowledge about the performance of the used wireless transmission channel along the route of the vehicle. In the following Chapter 6 we present several contributions, which address this problem for the cellular network.



(a) Start



(b) Middle



(c) End

Figure 35: Sections of the investigated construction site (yellow) (5th May 2016) in correlation with the situation after the completion (cyan) [89]. Satellite images ©Google Earth

6

OPTIMIZED DATA PROVISION FOR HIGHLY AUTOMATED DRIVING

In this Chapter we address our third research challenge (see Section 1.2): the reliable provisioning of data related to the maintenance of the HD Map via wireless cellular networks. Besides ad hoc communication technologies as described in the Sections 3.1.2 and 4.2, self-driving cars also rely on infrastructure-based wireless communication, such as the cellular network, to keep their HD Map operational. Due to the movement of the vehicle this connection is continuously affected by various influence factors. As illustrated by Figure 2 this includes static factors such as holes in the network coverage due to the lack of deployed cell towers and other geographic influences like mountains, woods or buildings. Dynamic influence factors include other network user, which share the same resources and weather conditions such as rain or fog that damp the cellular signal. All these influence factors lead to quickly changing levels of Quality of Service along the vehicle's route and make the robust transmission planning and the exchange of data challenging tasks.

Our contributions to meet these challenges, explained in the subsequent sections, are as follows:

i) Through our work in the course of the German research project Ko-HAF for self-driving vehicles, we are one of the first to provide insights on the actual performance requirements of self-driving vehicle's communication. Therefore an investigation of the present vehicular data streams is correlated with the cellular networks performance in a requirements analysis.

ii) Additionally an extensive measurement campaign is conducted to obtain actual performance data of the currently deployed cellular network infrastructure. Therefore the campaign covers the area around Frankfurt am Main, Germany, which includes all major highways and additional federal roads.

iii) We further investigated the creation of maps containing geo referenced key performance indicators of the cellular network. We propose to use these so called "Connectivity Maps" as an additional layer in the HD Map to be shared between all the self-driving cars. That way the self-driving vehicles obtain meaningful information regarding the network quality to be experienced in future. Based on this knowledge areas of poor network performance and connection losses for example can be avoided by preloading necessary HD Map data in advance. As our major contribution we develop and evaluate a framework (ICCOMQS) to cost- and time-efficiently generate such Connectivity Maps. Therefore we leverage the mandatory data transmissions of the self-driving vehicles (download of map updates, upload of vehicular sensor data) to probe the cellular network, without requiring any additional dummy data to be transferred. Consequently we save upon additional transmission costs, which are otherwise present in similar active probing approaches as mentioned in the Related Work (Sec. 3.3.2).

iv) As fourth contribution we improve the performance of the so called online prediction (see Section 3.3.3), an additional technique to support the management of the self-driving car's data transmission with special focus on the currently experienced network quality. Therefore we design and implement an Android smartphone application, the so called "Connectivity Map Client" to collect additional performance indicators of the cellular network infrastructure. From this dataset we derive specific sets of geographically limited training data to train a machine learning based predictor for the achievable throughput along the track. The algorithms prediction performance outperforms common approaches, which only use a single training data set.

*v)*Furthermore we develop and evaluate simulation frameworks, which are configured based on the obtained measurement data or further public available data sets. That way we enable the creation of accurate and scalable communication simulation scenarios. Compared to currently existing frameworks we achieve this without the high initial learning effort to develop a deep understanding of the necessary configuration process.

In the following Section 6.1 we perform our initial communication requirements analysis. Therefore we first identify the necessary amounts of data to be exchanged between the self-driving vehicles and the central backend entity, that hosts the HD Map, based on the requirements of a fleet of actual prototype vehicles. Subsequently we correlate those requirements with the theoretical and practical network capacities currently deployed on German highways.

6.1 COMMUNICATION REQUIREMENTS OF HIGHLY AUTOMATED VEHICLES

The necessary amount of data to be exchanged to successfully operate a HD Map is subject of current research [17, 18, 136, 177, 178]. To produce HD Map updates, various companies and research teams are using different vehicular sensor sources, which in conclusion produce varying amounts of raw sensor data. For this data to be actually used for the maintenance of the HD Map it needs to be transmitted to a central entity in the backend where further processing of the data is taking place. To derive meaningful knowledge from the raw sensor data to generate those updates is a subsequent challenging research task at the moment [18, 136].

Through our own research [92] conducted in the Ko-HAF project, to the best of our knowledge we are one of the first, to provide evaluation results regarding the communication requirements to operate a HD Map, based on an actual prototype implementation and extensive field tests as explained in the following Sections 6.1.1ff..

6.1.1 *Investigated Roadway Scenarios*

For our investigation we could rely on HD Map data for two different interconnected test areas as illustrated by Figure 36. The full combined track length thereby reached

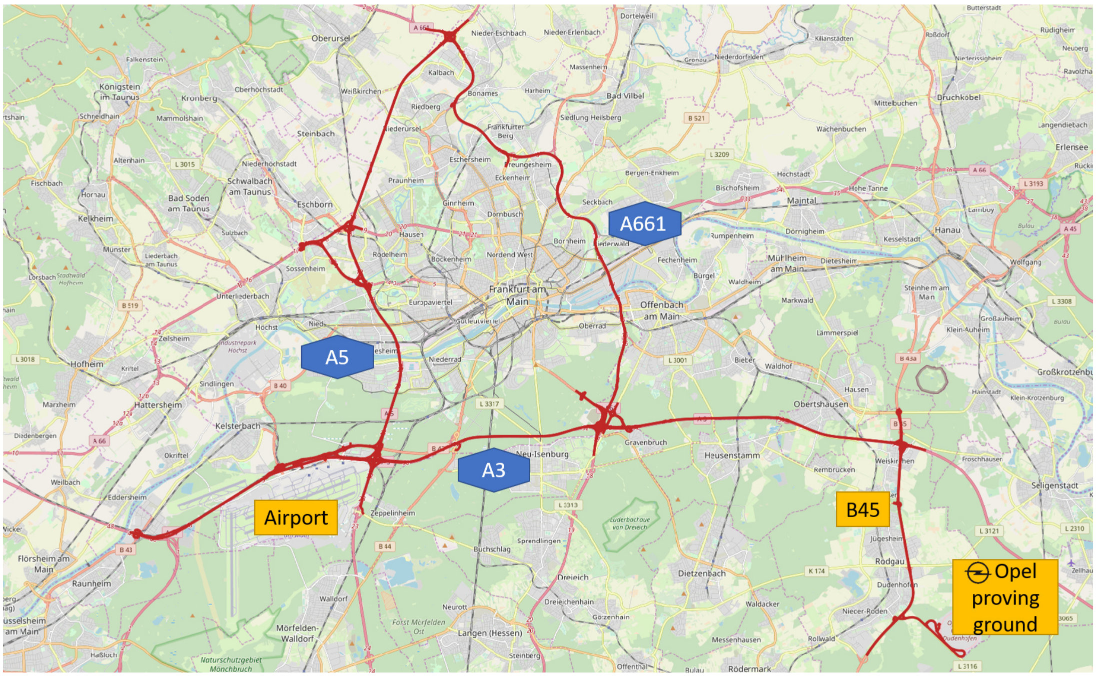


Figure 36: Overview of the complete Ko-HAF testing area. It includes sections of three motorways around Frankfurt am Main, Germany (A3, A5 and A661), the federal highway B45 and the Opel proving ground in Rodgau-Dudenhofen. ©Ko-HAF, 3D Mapping Solutions GmbH, Map data ©OpenStreetMap Contributors

about 275 km. This includes sections of the motorways A3, A5 and A661 and the federal highway B45 around Frankfurt am Main, Germany, as indicated by Figure 36. That way the fleet of self-driving prototype vehicles could operate in a challenging busy traffic environment with high quotas of commuting traffic. Additionally the closed off areal of the Opel proving ground in Rodgau-Dudenhofen allowed the self-driving vehicles to evaluate advanced and otherwise dangerous driving scenarios. This includes for example the detection and the safe bypass of lost cargo on the road. The testing ground as shown in the Figure 37 is a 4 kilometer track, resembling a four lane highway, with possible traveling speeds of up to 130 km/h.

Fleet of twelve self-driving prototype vehicles

6.1.2 *HD Map Data Download and Sensor Data Upload*

As a common concept explained in the Related Work (see Section 3.1) the full HD Map data was split up into smaller parts, so called map tiles, to be provided individually to the vehicles. The area that is covered by each map tile is one square angular minute, which is about 2.20 km^2 $(1,190m * 1.850m)$[20] in the considered geographic region of Germany. Depending on the complexity of the contained map data, the size of each individual map tile is varying. The individual sizes per map tile range from 12 KByte up to 373 KByte. The largest map tile, as shown in Figure 38, represents

---

[19] http://auto-presse.de/newssys/galerie/380981/380981_8_1000x.jpg (Last accessed on August 1, 2019)

[20] http://www.iaktueller.de/exx.php (Last accessed on August 1, 2019)

Figure 37: Aerial image of the highway section on the Opel proving ground in Rodgau-Dudenhofen. ©Opel Automobile GmbH[19]

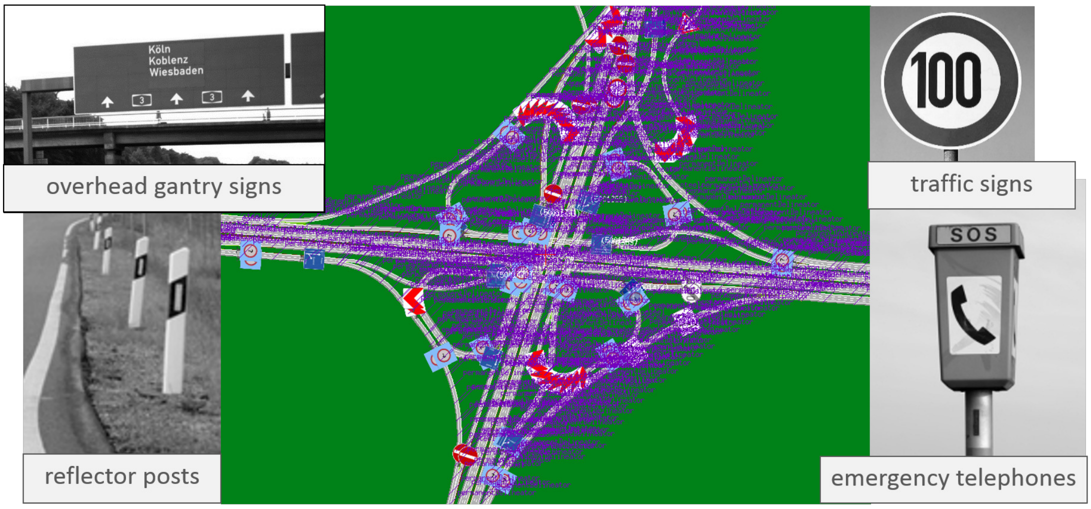
the complex situation at the "Frankfurter Kreuz" the major intersection between the highway A3 and A5 near the Frankfurt Rhein Main airport. As the available HD Map data specifically covers highway like streets, these data quantities are expected to be even larger in more complex areas, such as inner city street networks. Considering the overall average of 66 kBytes of data for a map tile requires a constant data downstream of approximately 55.5 kByte/km per vehicle, assuming a vehicle driving in parallel to the meridian of the earth and thus traveling about 1.19 km in one map tile (for calculation details see Section A.9 in the Appendix).

Amount of uploaded sensor data depended on equipment

The amount of sensor data that is uploaded from the self-driving vehicles to the HD Map maintaining server, heavily depends on their available sensory equipment and the reported objects. In our analysis the self-driving vehicles reported objects including: street signs, boundary lines, lane markings and reflector posts (see Figure 38b). The amount of used data to describe such an object also heavily varies with the achieved sampling resolutions of the sensory equipment of the vehicle. The figures 39a and 39b for example show two different examples of aggregated sensor samples of lane lines, with a varying sampling rate. Each measured sampling point is represented by 90 Bytes of data in the SENSORIS data exchange format [179, 180]. As consequence the different vehicles, produced a varying amount of sensor data per driven kilometer. The vehicle of Figure 39a produced between 170 - 252 kByte of sensor data per km, whereas the vehicle of Figure 39b collected data in the range of 578 - 1035 kByte per kilometer. This situation of varying data rates is expected to become even more diverse in future with the further development and research in the areas of sensor technology and data aggregation techniques.

In summary the required amount of data to be uploaded to the server is larger than the required amount to be downloaded. This is due to the fact, that the map changes

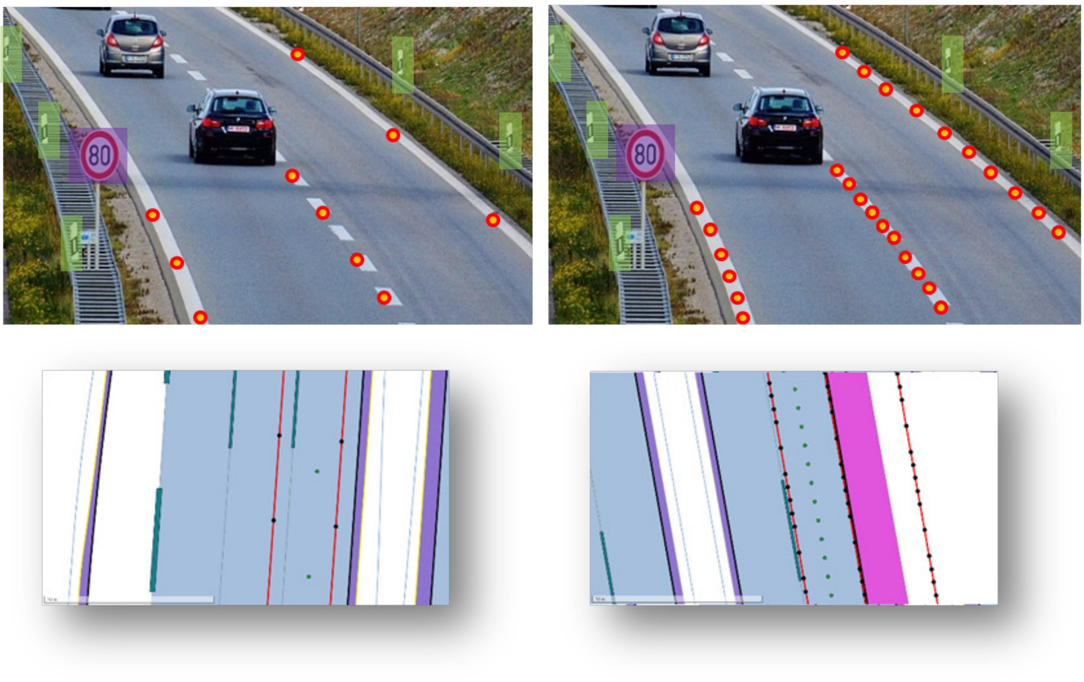
(a) Overview about the ”Frankfurter Kreuz” intersection between the highways A3 and A5 near the Frankfurt Rhein Main airport generated from Lidar Data.



(b) Map representation of the ”Frankfurter Kreuz” intersection, with all the associated objects along the track.

Figure 38: Visualization of the high degree of information contained in a single map tile of the HD Map. ©3D Mapping Solutions GmbH, Ko-HAF

can be described rather data efficient by only providing the changed information to the self-driving vehicles. The uploaded data however cannot be pre-filtered in the same way by the vehicle itself. For example a self-driving vehicle could overlook a still existing traffic sign, due to its occlusion by a truck nearby. Thus the map maintaining central server requires a constant verification of the correctness of the HD Map data, to detect and discard such false negative detections. Otherwise a several times not



(a) low sampling rate (250 kByte/km)

(b) high sampling rate (1035 kByte/km)

Figure 39: Varying data rates according to the sampling rates used by the different self-driving protoype vehicles to detect objects and landmarks [92]. Sample data ©Ko-HAF, photo source[21]

detected street element in the map data could then over time lead to a false update information from the server.

6.1.3 *Throughput Capacity Analysis*

Having derived the data requirements for the HD Map of a prototypical self-driving vehicle we now perform an initial analysis of the transmission capacities of the currently (2019) deployed cellular network infrastructure in the designated test area (Figure 36). Therefore we investigate an example scenario of one cell tower in the following. In terms of the operational functionality of the HD Map, as the main focus of our work, the available upload and download bandwidth of the wireless connection as well as it's predictable availability for necessary data transmissions are highly important. As illustrated by Figure 40 the identified data requirements for the HD Map in the upload direction is significantly higher than in the download direction. This is in contrast to the capabilities of the cellular network [87, 181], which provides higher available data throughput capacities in the downlink, than in the uplink. This holds true for all currently deployed cellular network technologies (e.g. LTE, HSPA, UMTS and EDGE) as well as for the currently specified future 5G network technologies. The

Capacity requirements higher in the upload

---

[21] http://tinytanksunblocked.club/mindestgeschwindigkeit-auf-deutschen-autobahnen/ (Last accessed on August 1, 2019)

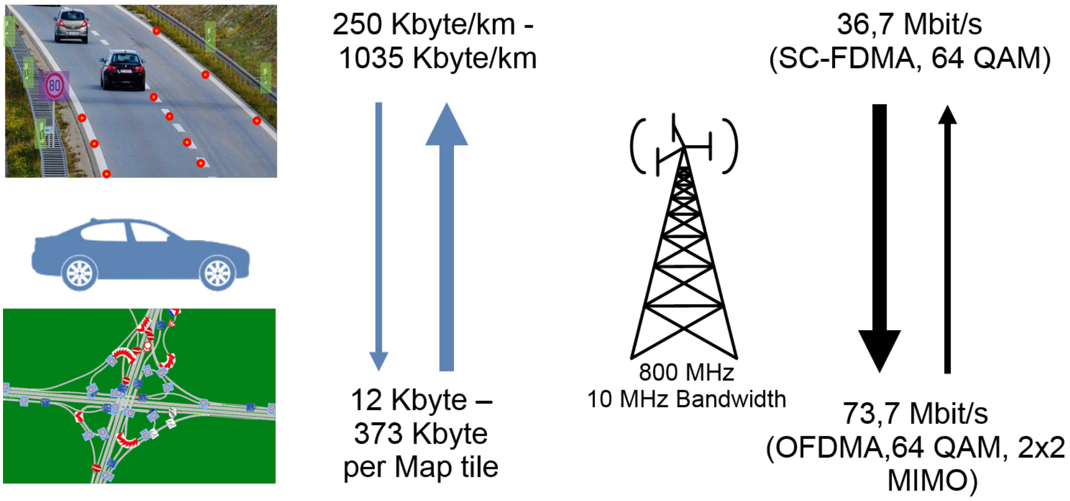


Figure 40: Comparison of the data requirements of one single self-driving prototype vehicle, used in the Ko-HAF project, and the data throughput capacities on the air interface of one example cell tower, as encountered in the test area. ©Ko-HAF, 3D Mapping Solutions GmbH, photo source[22]

reasons therefore include a reasonable and efficient power management for mobile, battery powered devices, such as smartphones (see Section A.3 for further background knowledge). LTE as the currently most advanced, deployed network technology for example uses a different multiple access scheme for the data transmission in the upload direction (Single Carrier Frequency Division Multiple Access - SC-FDMA) and in the download direction (Orthogonal Frequency Division Multiple Access - OFDMA). SC-FDMA in comparison to OFDMA is a more robust multiple access scheme, which is able to realize a stable transmission connection with less transmission power and thus less (battery) energy required. As a trade-off therefore SC-FDMA achieves a lower peak-data-throughput in comparison to OFDMA. The energy efficient data transmission on the client site (e.g smartphones) is further supported by the specification of the LTE frequency bands (see Table 2, 3 and 4 in [182]). Most of the defined operating bands have their uplink band in a lower frequency range, than their downlink band, as the lower frequencies possess better signal propagation properties, which allow the client device to use comparable less (battery) energy for the transmission. This also holds true for our own obtained measurements of the cellular network around Frankfurt (see Tables 19 and 20 in Section A.8 for more details). Only for the experienced operating band 20 this correlation is inversed.

To investigate the impact of these general correlations on the achievable throughput, Figure 40 illustrates an example calculation of the theoretical achievable peak bandwidths for a typical communication scenario as experienced by the self-driving vehicles in the test area around Frankfurt am Main. For the required background

---

[22] http://tinytanksunblocked.club/mindestgeschwindigkeit-auf-deutschen-autobahnen/ (Last accessed on August 1, 2019)

Example cell tower uses LTE Band 20 and 10 MHz bandwidth

knowledge on LTE see Section A.3. For the example scenario we assume one supplying cell tower with a transmission frequency in the range of 800 MHz (LTE Band 20) and a maximum available frequency bandwidth of ten MHz (as owned by all three German cellular network providers in that area of spectrum[23] at the time of writing this thesis). Furthermore we assume a perfect signal reception between the cell tower and the self-driving vehicle, as our receiving client, which enables them to use the highest Modulation and Coding Scheme (MCS) for their transmission. In correlation to the used vehicular communication hardware (see Figure 73), which contains a LTE modem of category 3 (LTE release 8) this allows them to exchange data in the downlink direction using the 64 QAM Modulation and Coding Scheme and a two antenna array on client side for a 2x2 Multiple Input Multiple Output (MIMO) data exchange. Assuming those network parameters and a full utilization of all its network resources (Packet Resource Blocks - PRBs) the cell tower achieves a theoretical peak download transmission speed of 73,7 Mbit/s (already excluding protocol overhead). In the opposite upload direction the self-driving vehicle can only achieve a transmission speed of 36.7 Mbit/s for the user data, using 16 QAM as the highest available Modulation and Coding Scheme in the uplink direction. Besides the different access schemes and Modulation and Coding schemes also the usage of only one transmission antenna in the upload direction (Single Input Single Output - SISO), due to a more efficient energy utilisation on the mobile client side, is a key reason therefore. The usage of only one single transmission antenna holds true for all devices categorized under the LTE releases 8 and 9. Within the LTE release 10, which is considered the first of the category LTE Advanced, the MIMO technology also has been introduced in the uplink direction (see page 15 in [182]). However the ratio of higher downlink and lower uplink speeds also holds true for such devices, too.

The assumed transmission frequency is reasonable for a vehicular communication scenario on a highway, as lower transmission frequencies correlate with a better signal propagation [183] and thus a larger coverage area. In the past the network providers tend to deploy cell towers with higher transmission frequencies and more available bandwidth (1800 MHz, LTE Band 3 and 2600 MHz, LTE Band 7[23]) [184, 185] mostly in urban areas, like big cities. With the providers focus on revenue those investments in cell towers with a correlating smaller coverage area, were only justified by the density of available paying customers. In conclusion their focus on revenue and also federal regulations [184, 186] steered the deployment of infrastructure into the direction of a large coverage of the German population, not a large geographic area as required for self-driving vehicles on the streets [185]. These literature based assumptions are further supported by our measured throughput values. 66.41 percent of all measurements where conducted via cell towers with a transmission frequency of 800 MHz and 33.59 percent in other frequencies as shown in Table 7. Recent proposed federal regulations [185] in Germany aim to improve this situation by imposing the cellular providers to deploy more infrastructure along the central highways and roads.

800 MHz transmission frequency experienced most often

We are well aware, that the identified data rates, which are required to maintain the HD Map of one single self-driving vehicle, should theoretically be easily achieved by

---

[23] https://www.spectrummonitoring.com/frequencies/#Germany (Last accessed on August 1, 2019)

Network Type	Range of Frequency [MHz]	No of Measurements	Percentage
UMTS	2100	348,534	9.78
LTE	800	2,365,948	66.41
LTE	1800	261,913	7.35
LTE	2100	128,034	9.78
LTE	2600	458,090	12.86

Table 7: Overview about the different technologies and frequency bands measured by the fleet of self-driving prototype vehicles. See tables 19 and 20 for an even more detailed listing.

modern day cellular technologies such as LTE. However these calculated maximum throughput capacity values have to be shared between all participating vehicles in one cell tower. In consequence our derived requirements towards the cellular network infrastructure capacities have to be scaled up for several hundreds of vehicles in parallel to correctly evaluate worst case scenarios, such as a morning rush hour. Furthermore we are well aware that technological improvements like Carrier Aggregation, more transmission antennas, higher modulation and coding schemes, higher transmission frequencies, ... (see Section A.3), raise the previously stated absolute throughput values. Compared to the vehicular communication hardware used during the Ko-HAF project such higher values can be achieved by more advanced cellular communication devices. However the calculated peak throughput values of LTE in our example scenario have to be put into the perspective of real world conditions. Several influencing factors such as the weather or the transmission distance between the self-driving vehicle and the cell tower, degrade the transmitted signals quality and thus lead to a reduction in transmission speed. Additionally the requirements regarding mobile data consumption are also expected to rise in future simultaneously [187–189], with the advancing development of the cellular technology. The self-driving vehicle itself, as a new data consumer and producer, enables further data consumption by its passengers, e.g. through streaming or video conferencing services. Those additional services thus also have to be taken into consideration, when evaluating the general data consumption of a self-driving vehicle [90].

The previously mentioned complex influencing factors cannot be easily incorporated into our initial analysis of the cellular network performance. Instead we performed an extensive measuring campaign of the cellular network to evaluate their impact. Therefore we relied on the fleet of twelve available prototype vehicles as measuring probes. Our obtained results for the highways around Frankfurt are described in the following Sections 6.1.4f.. The similar results for the Opel proving ground in Rodgau-Dudenhofen are described in Section A.10.

driven test kilometers	8435 km
period of conducted measuring campaign	2nd November 2017 - 17th November 2018
pure driving time	243 hours
amount of cellular data traffic used to probe the network	
upload	163.44 gigabyte
download	330.12 gigabyte
total	493.56 gigabyte

Table 8: General information about the cellular network measurements performed during the test drives in the Ko-HAF project.

6.1.4 *Measuring Campaign - Setup description*

Vehicular communication device of LTE category 3 used for campaign

All vehicles were equipped with the same set of transmission antennas as well as communication hardware (COM-Box), which allowed us to compare all conducted measurements with each other, without the necessity to consider various hardware influences. The COM-Box hosted an LTE modem of the category 3. Thus the theoretical peak throughput speeds, that the box could achieve were 100 Mbit/s in the downlink and 50 Mbit/s in the uplink. The cellular measurements were conducted through additionally transmitted probing packets (the so called dummy data), strictly separated from the actual necessary data transmissions required to exchange HD Map updates and vehicular sensor data. That way we could ensure that the gained measuring results were not influenced by different patterns of underlying data traffic on the client side. To conduct the throughput estimations of the cellular network, data chunks of four megabytes in size were downloaded and uploaded via the HTTP protocol, relying on the TCP protocol as reliable transmission protocol. The data amount of four megabytes was selected based on extensive initial test measurements to ensure that the obtained measuring results were not influenced through insufficient resource allocations from the serving cell tower or influences from the TCP transport layer protocol (e.g. the initial slow start phase at the beginning of a data transmission). To ensure that the map servers networking capacities were not creating a bottleneck during our throughput estimation we limited the amount of possible conductible test to only one at a time, by limiting the amount of possible HTTP requests to one. Further incoming requests by the other cars during the execution phase of the estimation were neglected.

Over one year long measuring campaign

As the cellular measurements were executed in parallel to the test drives, we achieved a run time of our measuring campaign spanning over nearly the full active testing phase of the Ko-HAF project. In the end cellular data could be collected over a period of more than one year of test drives (2nd November 2017 - 17th November 2018 - see Table 8). A total track length of 8435 km has been covered during a period of 243 hours of pure driving time. During this time a total of 493.56 gigabyte

measurable through provided interfaces	cellular network type, Reference Signal Received Power (RSRP), Received Signal Strength Indicaton (RSSI), Signal to Noise Ratio (SINR), Reference Signal Received Quality (RSRQ), cell ID, cell frequency
measurable through active data transmissions	upload and download throughput [Mbit/s], Round Trip Time (RTT) [ms]
additional data	vehicle ID, vehicle position, vehicle speed, timestamp

Table 9: Collected communication-related parameters from the Ko-HAF vehicle fleet.

of cellular data have been used to conduct throughput estimations in the uplink and the downlink direction.

To describe the self-driving vehicles experienced network quality comprehensively, several further key network performance parameters have been captured through the COM-Box (see Table 9). Besides the data throughput measurements also latency estimations of the current connection have been actively measured. Therefore we executed ping measurements, by measuring the required time for the TCP connection establishment, via the SYN/ACK packets, as well as the tear down of the TCP connection at the end of the data transmission for the active throughput probing, via the FIN/ACK packets. Additionally the Ko-HAF COM-Box provided several further performance indicators through its available interfaces. Those values did not require the active transmission of data over the cellular channel. This for example included important performance indicators regarding the currently available signal strength (e.g. the Reference Signal Received Power (RSRP), the Received Signal Strength Indicaton (RSSI), the Signal to Noise Ratio (SINR) and the derived Reference Signal Received Quality (RSRQ) value. Furthermore to identify the different serving cell towers, along the way around the test tracks, their correlating cell ID and their individual transmission frequency was logged. All these collected values are stored in a large database as a geo referenced data set that can be correlated with the vehicles position, its speed and the general time, when each measurement was conducted. This geo referenced database represents the "Connectivity Map" of the test area as introduced by the Related Work of Section 3.3.1. It allows us to investigate the experienced network quality and available infrastructure as described in the following Section 6.1.5.

6.1.5 *Measuring Campaign - Frankfurt*

Figure 41 summarizes the achieved mean of all four major key performance indicators in the test area. They are throughput in the uplink and downlink, as well as the Round Trip Time (RTT) and the Reference Signal Received Power (RSRP). Each dot on the map thereby represents the aggregated result of a circular segment of 500 meters in diameter on the test track. The color of the dot thereby represents the mean of all the measurements conducted within its range (from green to red / good to poor performance). The mean upload and download throughput are shown in Figure 41a and 41b, whereas the RTT and the RSRP are represented in Figure 41c and 41d respectively.

Several defined areas of high and low performance

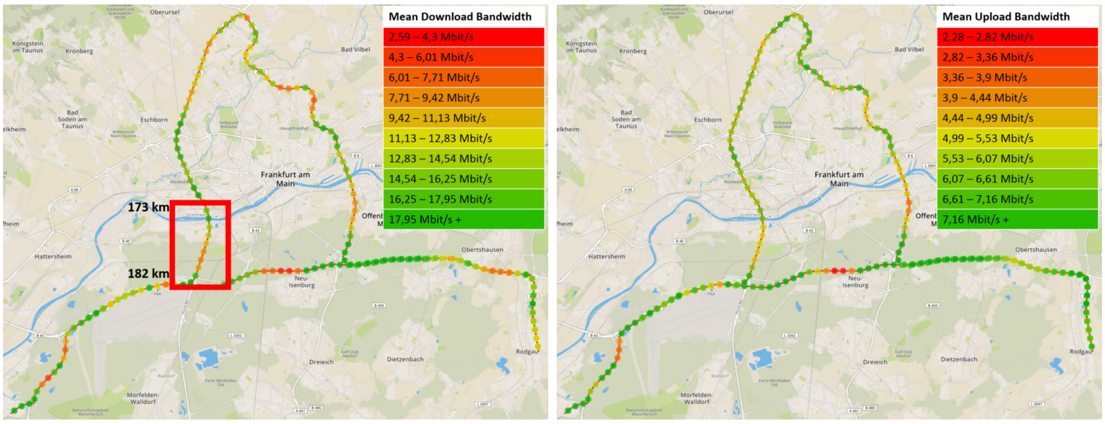
The gained measuring results clearly showcase real-world examples for our initial statement of varying geographic cellular network performance as explained in Section 1.2. Along the investigated highways exist several areas with significantly higher or lower achieved performance values of the four indicators compared to the overall average of the complete map. Similar results were obtained for the proving ground in Rodgau-Dudenhofen as shown in Section A.10. These geographically referenced, rather good or poor areas for a possible data transmission justify the development of further techniques to improve the overall user-experienced transmission performance as described in the following Sections 6.2ff..

In the subsequent evaluation step, explained in the following, we put our gained measurement results into the context of an actual deployment scenario of self-driving vehicles in daily traffic.

Correlation with Daily Vehicular Traffic

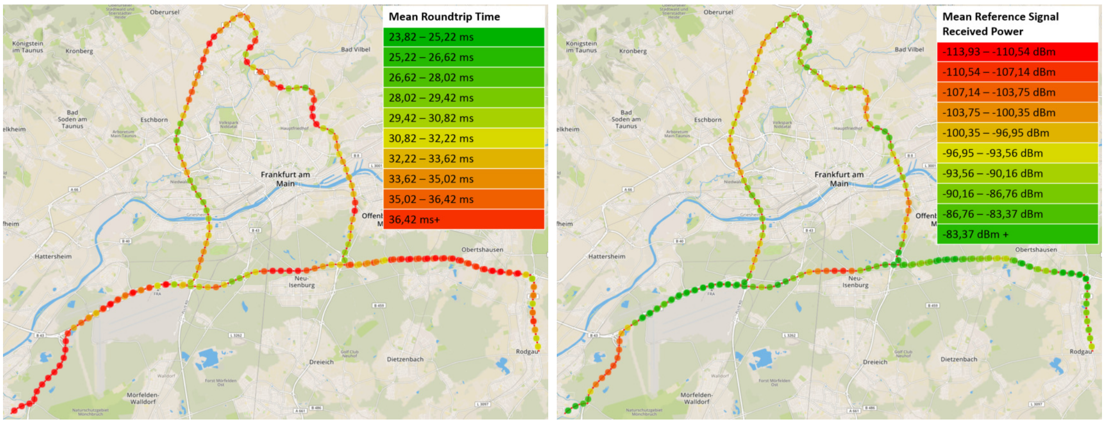
Peak of vehicular traffic in morning rush hour

To investigate a possible worst case communication scenario we obtained detailed information about the daily average vehicular traffic on the highways in our test area from the German Federal Road Research Institute (BASt) [190]. The area with the highest vehicular traffic is located on the motorway A5 near the off ramp of Frankfurt-Niederrad as indicated by the red rectangle of Figure 41a. Due to the dense commuting traffic into and out of Frankfurt an average of 156,000 vehicles roams in this area on a daily basis. To identify the time of the day with the highest possible load upon the cellular network we investigated the hourly averages of vehicular traffic. As expected the vehicular traffic peaks in the morning and evening rush hours (8-9 o'clock and 17-18 o'clock). The highest number of considerable mobile clients in the cellular network is achieved around 8 o'clock in the morning with 15,158 vehicles/per hour crossing the area. To put these numbers in correlation to a detailed perspective of the networks capacity the boxplots of the Figures 42 and 43 visualize the measured performance values (throughput, RTT and RSRP) of the cellular network along the aggregated segments of the track. Most important for our evaluation are the achieved upload and download throughput values, as indicated in Figure 42a. Each boxplot thereby indicates the distribution of all measured values in the perimeter of the specified road kilometer along the x-axis (see Figure 41a for comparison). For this specific scenario comparable higher data rates at the beginning and at the end of the track correlate with higher measured RSRP values. An evaluation result which showcases the impact of the different Modulation and Coding Schemes that are applied based on the received signal strength (as explained in Sec. 6.1.3). However a sole measuring of the RSRP value is not sufficient enough to derive the network's throughput performance. Both values have to be collected individually. In terms of the throughput this consequently requires active probing of the network. Figure 44 shows two example locations that verify this statement, as there is no direct correlation between the throughput and the RSRP value visible. The first location is nearby the Frankfurt Rhein Main airport located in



(a) Visualized mean average download bandwidth of the provider Vodafone as derived from the measurements of the fleet of probe vehicles.

(b) Visualized mean average upload bandwidth of the provider Vodafone as derived from the measurements of the fleet of probe vehicles.



(c) Visualized mean average round trip time of the provider Vodafone as derived from the measurements of the fleet of probe vehicles.

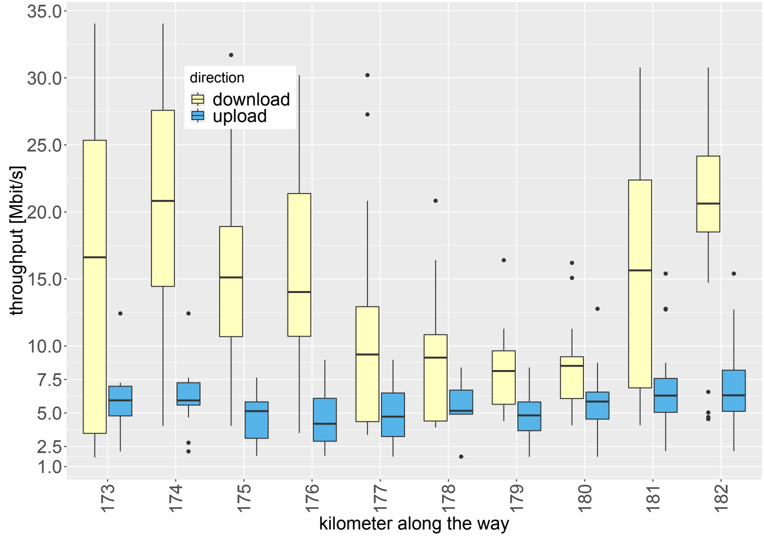
(d) Visualized mean average reference signal received power of the provider Vodafone as derived from the measurements of the fleet of probe vehicles.

Figure 41: Overview about the different measured key performance indicators of the LTE network on the Ko-HAF testing area around Frankfurt [92]. Map data ©OpenStreetMap contributors

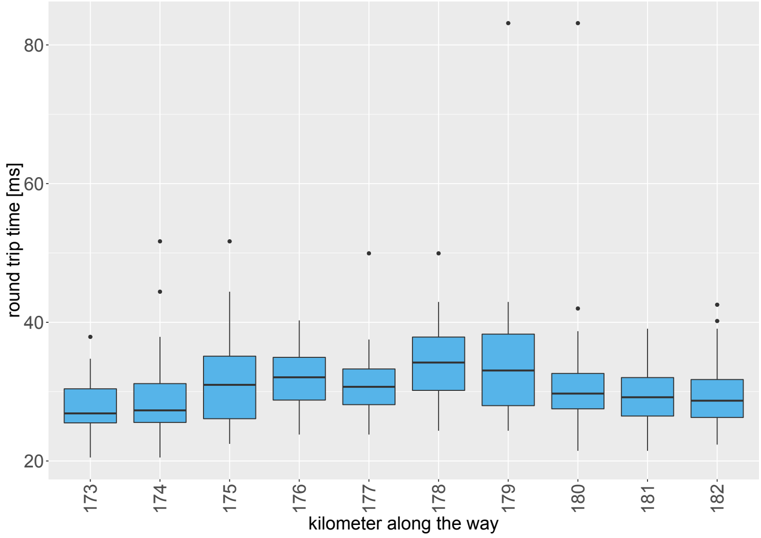
the southwest of the area. The other location is nearby the city of Obertshausen in the east of the testing area. As stated in Section 1.2 the achieved throughput values are influenced by many different factors. For the area around the Frankfurt Rhein Main airport probably a dense active user basis shows its impact on the LTE network as a shared medium.

In comparison to the theoretical possible throughput values as calculated for one exemplary cell tower in Section 6.1.3 the actual achieved throughput values in the test area are considerably lower. This clearly shows the significant negative impact on the cellular connection that results from the various influencing factors in an actual real world environment. For the download direction the majority of the achieved speed

Measured throughput values are significantly lower than theoretical peak values



(a) Measured upload and download throughput in the investigated road section on the highway A5.

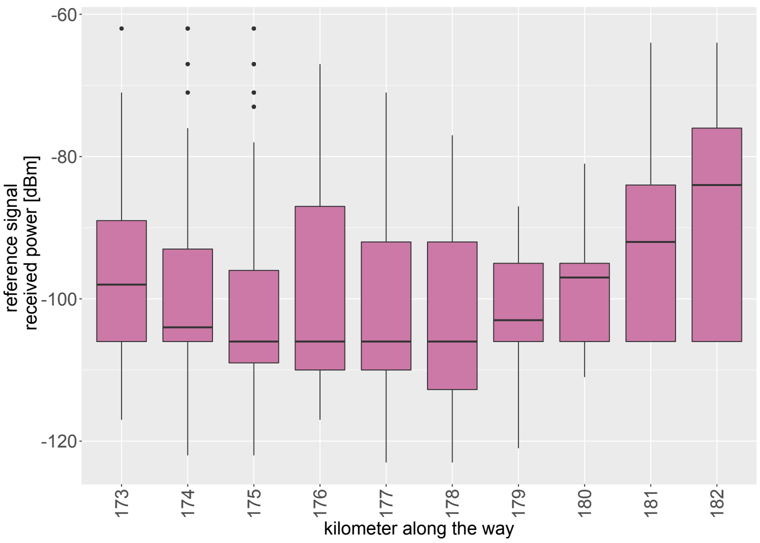


(b) Measured Round trip times in the investigated road section on the highway A5.

Figure 42: Overview about the different measured key performance indicators of the LTE network on the designated section on the motorway A5 near Frankfurt-Niederrad [92].

values largely varies between values as high as 35 Mbit/s and as low as only 1.75 Mbit/s. In correlation to our statements of Section 6.1.4 the achieved upload speeds reach even smaller values between 12.5 Mbit/s and 1.75 Mbit/s. Especially these large variances in the achieved throughputs have to be considered for a robust HD Map update process.

To be able to put the daily traffic and the achieved data rates into perspective to each other we furthermore had to investigate the deployed cellular network infrastructure



(a) Achieved Reference Signal Received Power (RSRP) in the investigated road section on the highway A5.

Figure 43: Continued overview about the different measured key performance indicators of the LTE network on the designated section on the motorway A5 near Frankfurt-Niederrad [92].

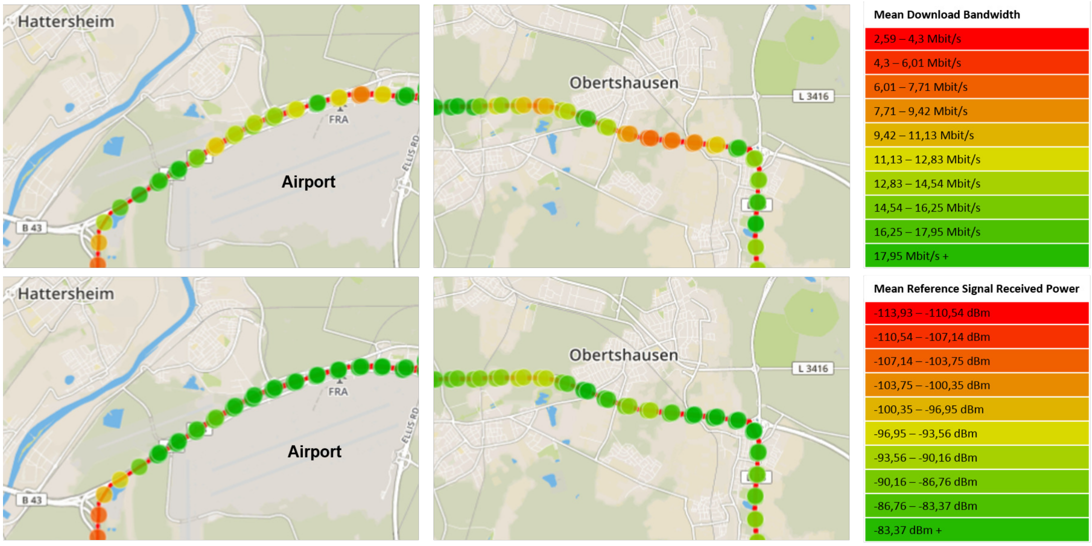


Figure 44: Comparison of the Reference Signal Received Power to the achieved download throughput in selected areas along the test track [92]. The indicated areas showcase, that a direct correlation between the received signal strength and the achieved throughput is not always present. Further influencing factors such as the number of active clients require active probing to obtain the cellular network's throughput. Map data ©OpenStreetMap contributors

in the marked area. Based on our measurement data we identified that at each point along the route near Frankfurt-Niederrad the investigated provider Vodafone had

Upload data rate each cell tower has to provide for:		Download data rate each cell tower has to provide for:
Vehicular sensor data upload at 250 kByte/km per vehicle	Vehicular sensor data upload at 1035 kByte/km per vehicle	Map data download at 55.5 kByte/km per vehicle
4.49 Mbit/s	18.59 Mbit/s	0.997 Mbit/s

Table 10: Mandatory network throughput capacity each cell tower has to provide for all self-driving vehicles in the worst case communication scenario near Frankfurt-Niederrad.

deployed two cell towers to provide its costumers. Furthermore we measured an average cell transmission range of 3.2 km. This transmission range is comparable to the values achieved for a similar urban scenario described in the Related Work of Afric and Pilinsky [191]. In their scenario the authors stated that the average achieved transmission range of an LTE cell tower was around 5.5 km. Assuming a similar deployed network infrastructure for the two remaining cellular network providers in Germany (Telekom and O2) at the time of the measuring campaign, we considered an amount of 6 cells in total to be available in our evaluation scenario to equally serve the present vehicles. Furthermore we assumed an equal timely distribution of the 15,158 vehicles passing through the considered area during the rush hour. The traveling speed of each vehicle was assumed to be 100 km/h as an average over the speeds of slower trucks (speed limit of 80 km/h [192]) and faster sedans (average speed of 125 km/h [193]). The upload and download data rates were set accordingly to the achieved values during the Ko-HAF project (two different upload data rates: i) 250 kByte/km and ii) 1035 kByte/km, as well as a download data rate of 55.5 kByte/km).

With all this information at hand we calculated the required upload and download network capacities that each of the cell towers had to full fill as worst case scenario. The achieved results are summarized in Table 10. For calculation details see Section A.9 in the Appendix. When comparing the calculated values with the measured upload and download throughput values of Figure 42a especially the required upload data rates show a critical impact on the available network resources. Even assuming only the lowest data rate to provide the vehicular sensor information already comes close to the average mean provided upload data rates of the network. Considering the scenario of a high upload data rate the available network resources are even exceeded. In comparison the necessary download data rates can be more easily served. However there are still certain critical segments along the investigated track between street-kilometer 173 and 182, where the gap between available resources and required data rates becomes very close. Additionally the download direction also has to be considered to serve further additional services with a significant impact, such as audio and video streaming in parallel, which further fortifies the load on the cellular network. For further details regarding their possible impact we refer to our related publication [92].

Vehicular sensor data upload has strong impact on cellular network resources

In conclusion we could identify that even with a very conservative data foot print, as required by the examined self-driving vehicles, significant additional traffic load is going to be introduced into the deployed cellular network to maintain the HD Map. This additional load is especially critical in the upload direction, as the cellular networks capacity is optimized for the provisioning of content in the download direction.

6.2 MEASUREMENT CAMPAIGN CONCLUSIONS - CONNECTIVITY MAP AS HD MAP LAYER

The cellular network measurements, conducted in the Frankfurt test area clearly visualize our initial statement of Chapter 6: The available quality of the cellular network is quickly changing along the route of the self-driving vehicle due to several external influencing factors according with the high mobility of the vehicle itself.

Based on our initial thoughts for this research area, we do not consider the Connectivity Map as some kind of external information source, as introduced in the Related Work. Instead we propose its integration into the HD Map, as an additional, distinct and dynamic layer, as shown in Figure 45. That way the maintenance application of the Connectivity Map in all self-driving vehicles can rely on the same interfaces and techniques that are already available to exchange the further traffic information located in the other layers of the HD Map. This ensures technological robustness and enables an easier future dissemination of the technology.

To enable the efficient maintenance of this distinct HD Map layer we propose a specialized framework as described in the following Section 6.3.

6.3 CONNECTIVITY MAP CREATION FRAMEWORK - ICCOMQS

The major contribution of the Intelligent Cellular Communication Quality Sensing framework (ICCOMQS) in contrast to all Related Work lies in the generation of the cellular probing data and its intelligent dispersion to efficiently cover a large geographic area. In contrast to the work of Papageorge et al. [154] we do not piggyback our sensor data upon dummy data packets for the probing process. As explained previously, this might in the worst case degrade down to the level of active measuring by sending nearly pure dummy data. The ICCOMQS framework instead only uses the data to be send anyway in mobile sensor networks as probing units. In contrast to a passive measuring approach ICCOMQS schedules the amount of data to be send for probing and its transmission time completely on its own. Thus it achieves the same performance results in terms of measuring accuracy as an active probing mechanism, without its additional costs. To achieve all these benefits ICCOMQS relies on the fundamental property of transmission delay tolerance for certain types of data to be exchanged between the sensor nodes and a data processing backend entity (e.g. a central server) as explained in the following. Obviously there exist different kinds of data types, with varying time requirements. Thus extremely time critical events such as accident warn-

Active probing measuring accuracy without additional costs

---

24 https://ko-haf.de/fileadmin/user_upload/media/abschlusspraesentation/14_Ko-HAF_Continuous-Updating-of-Backend-HD-Map-Data.pdf (Last accessed on August 1, 2019)

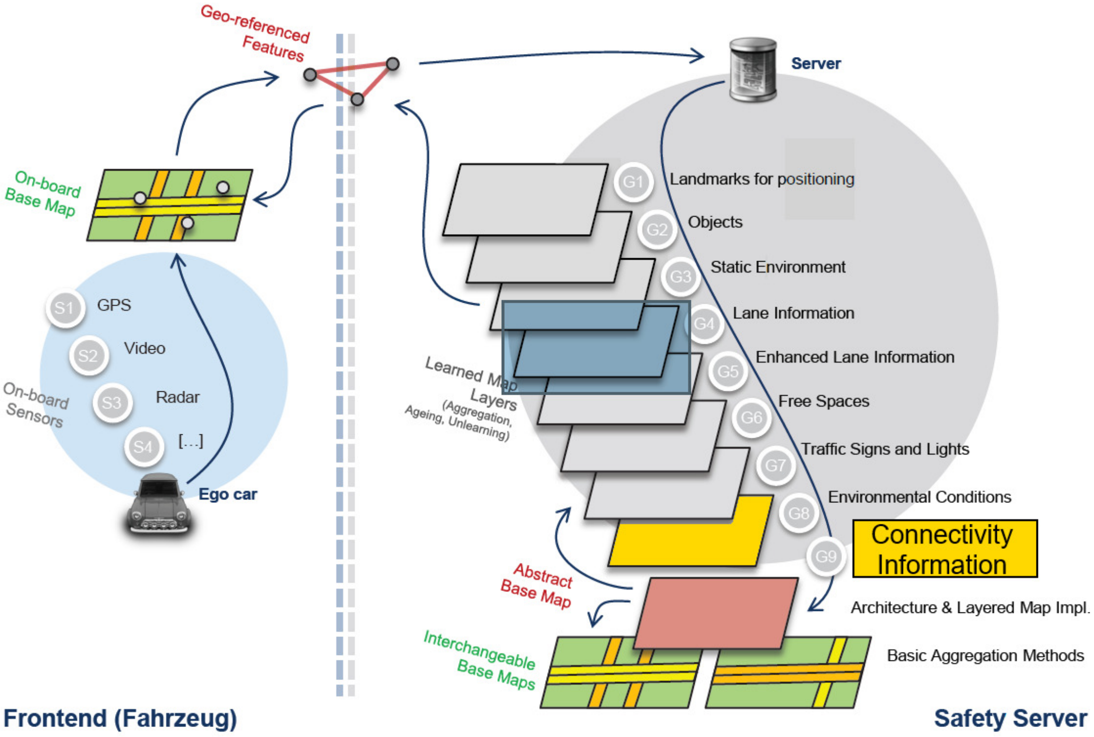


Figure 45: Proposed integration of the Connectivity Map as an important additional layer of dynamic information into the HD Map for the self-driving vehicles. The layers information can be shared similar to the traffic information on the other layers, ensuring robustness in the deployment. ©Ko-HAF[24]

ings ahead on the road have to be excluded from the following sketched scheduling procedure and instead be sent right away. The remaining data, however posses a certain time frame (delay tolerance) of minutes to hours to be collected or processed to achieve reasonable information quality in the backend to be used in the subsequent applications. This condition allows ICCOMQS to delay the transmission of this kind of data, within its given time constraints, on client or server side as illustrated by Figure 46. ICCOMQS then schedules the transmission of the useful data into geographic areas where information about the cellular network quality shall be collected, without introducing any additional transmission costs. The sensor data provided by a self-driving vehicle to a central data processing backend server and the received HD Map updates are a key example for such a kind of delay tolerant data. To be able to provide a reliable map update, the data processing algorithms in the backend first have to receive enough sensor data on the changed traffic situation from the vehicles. How much sensor data actually is required to successfully maintain a HD Map, depends on the information that needs to be updated and is still subject of ongoing research. Klejnowski et al. [136] for example investigated this question for the update use case of changed lane markings. In the work of Beringer et al. [18] the authors describe a system (Elektrobit robinos) to aggregate sensor data and to provide HD Map updates based on this sensory information. Elektrobit robinos thereby provides updates of

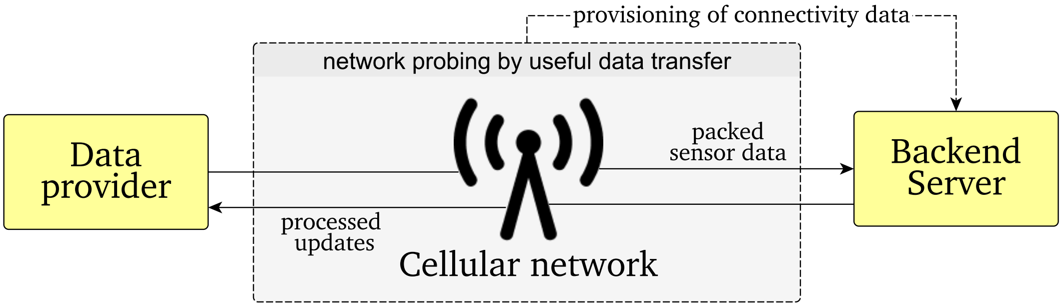


Figure 46: General working principle of the ICCOMQS framework [93].

the map data on a daily basis. A similar degree of delay tolerance is also given for the provisioning of the finalized map update to the self-driving vehicle. The car itself has to receive the data before it reaches the certain geographic area of its relevance (see Section 4.1). This gives the ICCOMQS framework's entity on the side of the data providing backend (see Figure 46) a certain time window, in which the transmission can be scheduled. The ICCOMQS framework in general abstracts from this described specific use case in the domain of self-driving vehicles. It was designed by us to be applied upon any mobile sensor network's data, which posses a similar property of transmission time tolerance.

The detailed functionality of the framework and the design decisions, which lead to its final composition are described in the following Section 6.3.1.

6.3.1 *Design Requirements Analysis*

Generalization and flexibility

Our major concern in the design and specification phase of ICCOMQS was to develop the framework, with the aforementioned key functionalities, without being limited to a single type of wireless network or a single type of exchangeable data in mind. Instead the framework had to be capable to operate with any given wireless network and data source, to ensure a reliable further usage by researchers or engineers in as many as possible usage scenarios. Consequently ICCOMQS has been developed with generalization and flexibility in mind. Figure 47 illustrates its detailed design concept for the envisioned practical scenario of a client sever architecture to probe the cellular network as example of a wireless network. However the only assumption we made to realize the framework's capabilities described in the previous Section 6.3, was that two network entities (indicated as client and server in the Figure 47) communicate through a wireless network (the cellular network in our usage scenario) with each other. The entities were envisioned to provide either sensor data (the client) or to store and process the data (the server) and forward an aggregated result back to the other communicating entities. To address the problem of diversity in the data to be exchanged, we decided to use Google's Protocol Buffers (protobuf[25]) as common data format (PB_*) to interface with any given data source on both communicating entities.

---

[25] https://developers.google.com/protocol-buffers/ (Last accessed on August 1, 2019)

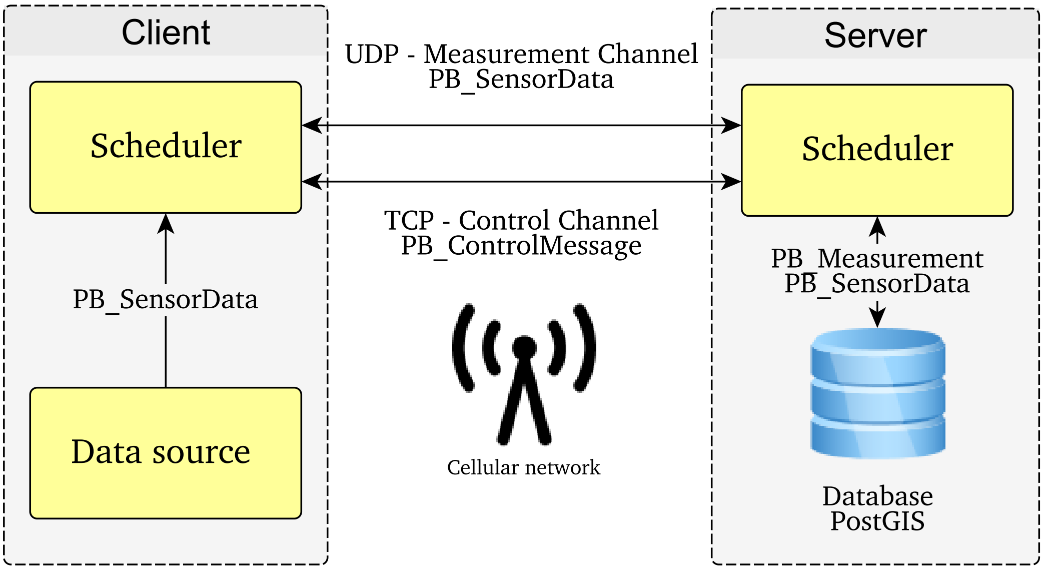


Figure 47: Design of the ICCOMQS framework [93].

Protobuf provides several key functionalities that led to this design decision. First of all the data format can contain any kind of data (simple data types as well as complex data objects and lists of them) and provides common interfaces for several different programming languages (e.g. C++, Java and Python), which get automatically generated from the formats specification files. This ensures robustness and high flexibility for the easy integration of existing subsequent applications. Additional data variables for example can be easily added without the necessity of manual altering all interface functions. The program interfaces are automatically updated by the protobuf compiler, when the common data format gets altered. As second design consequence we could not assume a certain operating platform on which the ICCOMQS software is executed. Furthermore we required fast and accurate timing performance of the framework itself to realize reliable active cellular measurements (e.g. latency and throughput estimations). As a result and to stay as close as possible to the used hardware's capabilities we decided to implement the framework in the programming language C++. C++ provides us with features such as object orientation and hardware optimized code execution for reliable timings during the measurements. Furthermore a variety of compilers is available to run the framework on nearly all existing operating platforms (e.g. Microsoft Windows, Linux, Mac OS) as well as different architectures (e.g. ARM and x86). To ensure the efficient operation of the ICCOMQS framework based only on the provided user data, we had to design its measuring process of the wireless network as data-efficient as possible. To achieve this goal, we did not rely upon the Transmission Control Protocol (TCP), as used in the Related work of Papageorge et al. [154]. TCP assumed to be used in one of its most common configurations inside the mobile nodes uses congestion control mechanisms. This includes techniques such as slow start and congestion avoidance. Thus it uses significant amounts of data in

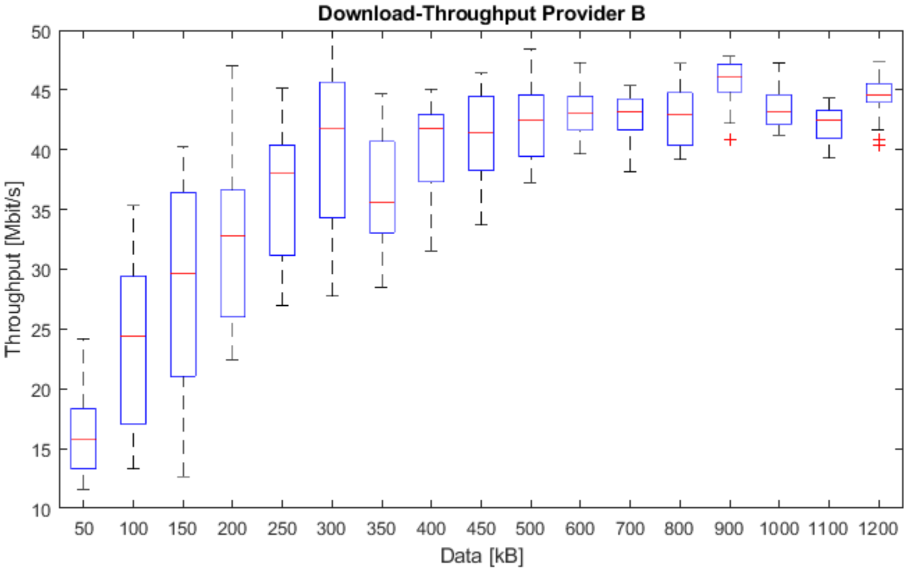


Figure 48: Exemplary investigation of the minimum amount of data required to probe an LTE celltower with a transmission bandwidth of 10 MHz (LTE Band 20 - 800 MHz). By using an LG Nexus 5 smartphone as measuring device the theoretical maximum throughput that can be achieved under perfect signal conditions, including the protocol overhead is 51.5 Mbit/s (50 resource blocks, QAM 64). At the position of the stationary measurement the connection speed saturates around 45 Mbit/s, close to the theoretical capacity limit, using only about 1 MByte of probing data.

the transmission process before reaching the network's peak bandwidth, which shall be included as key performance indicator in the Connectivity Map. Similar to most of the presented active probing algorithms of Section 3.3.1f. we relied upon the User Datagram Protocol (UDP) instead, as it gives us the full control about the pattern of data to be send over the network to reach its capacity limit as quickly as possible. Xu et al. [153] especially have shown the capabilities of UDP to reliably probe the limits of modern cellular networks with only a very small amount of available probing data (only several kilobytes to some megabytes). The authors thereby could not identify any difference (e.g. introduced by artificial regulation rules of the cellular providers) in the obtained final measuring results when using TCP or UDP.

UDP for probing

To ensure the application of these results upon our personal network infrastructure we conducted own performance tests in our investigated LTE infrastructure with similar results. Figure 48 exemplary illustrates the amount of data required to probe a cell tower with a transmission bandwidth of 10 MHz or 50 resource blocks respectively (see Sec. A.3 for details). For the probing process we used a LG Nexus 5 smartphone, which is capable of using 64 QAM as highest modulation and coding scheme, but no MIMO technology. In summary the theoretical throughput speed limit of this device being served by the cell tower is 51.5 Mbit/s in the downlink (including the protocol overhead). At the site where the stationary measurements were conducted we could identify a saturation of the throughput estimation at around 45 Mbit/s, which could be achieved by using about 1 MByte of probing data and is close to the cells capacity limit considering normal signal conditions. The cell site was hosted by provider B as referred to in the following Section 6.4.1, one of the three providers hosting cell sites in Germany at the time of writing this thesis (2019). Similar results were also obtained for the other two providers.

However ICCOMQS can be easily adapted to use TCP instead of UDP for the probing mechanism, if the achieved measuring results should differ for service providers in other countries or the established flow control mechanisms on the provider's side should change in future. In the final configuration and the prototypical implementation of the ICCOMQS in an Android client (see Section 6.4.1 for further details) we used an adaptive measuring algorithm, which increased and decreased the amount of sensor data used for the cellular probing to stay as close as possible to the networks transmission limits. In contrast to the techniques used in TCP, we thereby relied on the directly measured network quality indicators (e.g. the currently available transmission bandwidth in Hertz), as well as the ones stored in the Connectivity Map to achieve a faster adaptation rate. Similar Quality of Service indicators can also be collected for other wireless transmission networks. Based on our personal conducted measurements (see the previous Section 6.1) we additionally agree with the further statement of Xu et al. [153], that measurements of the cellular network should be conducted as quickly as possible, not to be affected by certain long term effects, such as the introduction of additional load on the channel introduced by further applications. To ensure a reliable transmission of the valuable user data via the unreliable UDP, we further introduced an additional TCP control channel. That way the client or server side can initiate the retransmission of lost packets if necessary. Instead of using two different transmission channels (one for the probing data and one for the control information) also the use of the QUIC protocol [194] is considered for future work, as it also relies on UDP on the transport layer of the network stack and provides further potential for network specific adaptation through the control mechanisms provided on the application layer.

6.3.2 *The Scheduler*

The key functionality of the ICCOMQS framework resides in its Scheduler component, which exists in the same fashion in all the communicating network entities, as illustrated by Figure 47 for the client and server. Its main purpose is to achieve the previously described geographic coverage through different scheduling strategies, explained in the following Section 6.3.3 as our second major contribution in the design and implementation of ICCOMQS . To enable these intelligent probing strategies the Scheduler, shown in detail in Figure 49, has to collect and store the incoming user data to be exchanged between the communication parties. Figure 49 illustrates all the designated steps of the necessary processing pipeline in the upload direction between a data source on client side and the backend database on the server side. The general principle to probe the wireless channel's performance however stays the same for both directions (upload and download). As the first step in the pipeline the Data collection subcomponent inside the Scheduler packs any incoming data accordingly to their tim-

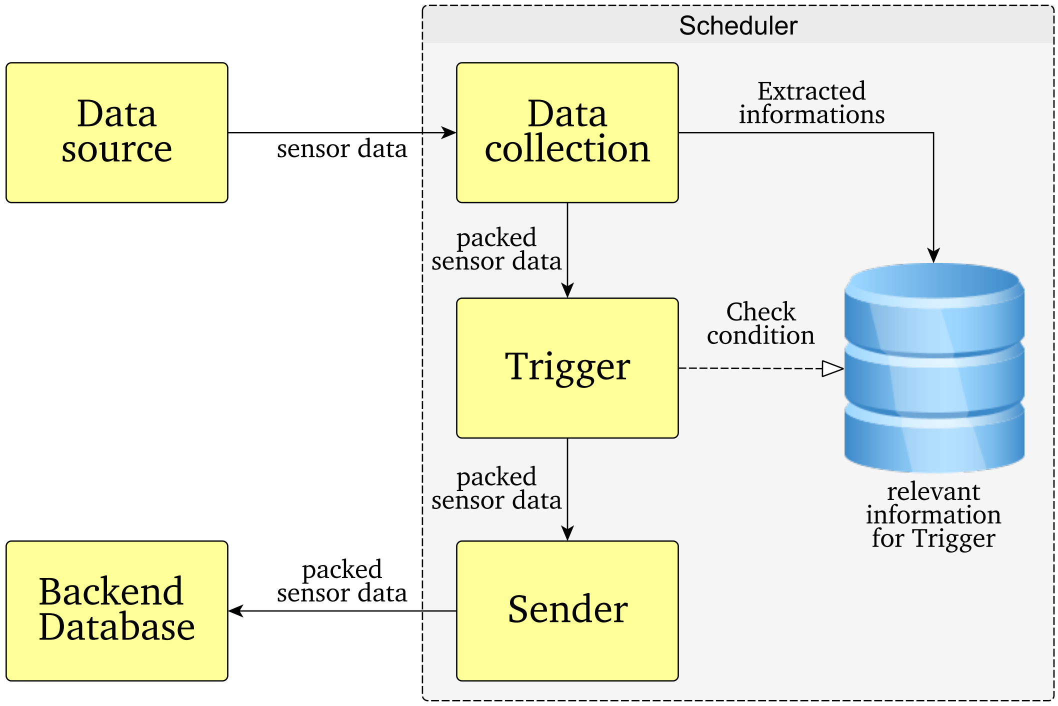


Figure 49: Processing chain within the scheduler module of the framework for the upload direction from client to server [93].

ing requirements together to fit into the desired size of a probing packet. That way the Data collection component packages data until the critical mass for a network measurement is reached and then informs the Trigger component as the following entity. It also extracts further relevant information (e.g. the measured geographic location) out of the user data to provide it to the Trigger. The Trigger component then plans on which position along the vehicles track the measurement of the wireless network is going to be executed. The chosen side conditions for this planning process can be configured freely, e.g. by the extracted information of the provided user data. For our probing algorithms as proposed in Section 6.3.3 and evaluated in Section 6.3.4ff. we only assumed to be provided with the localization information of the mobile nodes.

6.3.3 *Intelligent Probing Strategies*

Our second major contribution in the context of the ICCOMQS framework is the design and the investigation of different probing strategies to improve the geographic dispersion of measurements of the cellular network to be stored in the Connectivity Map. To the best of our knowledge we were the first to focus on this kind of aspect within the creation of the Connectivity Map, as all other investigated works (see Sections 3.3.1 and 3.3.2) either had a different focus on the dispersion of network tests or relied on different prerequisites for the probing procedure. In the following we discuss the four different probing strategies, which we designed for the application in

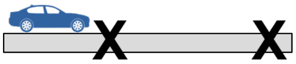
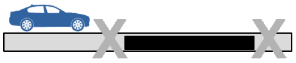
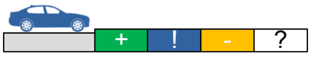
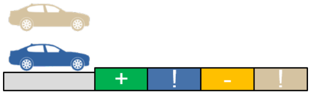
Approach	Time of Measurement	Example
Trivial	Executed as soon as the data for one measurement is ready.	
Aimed Random	Executed randomly in the time frame between two possible trivial measurements.	
Local Map	Planned accordingly to the white spots in the connectivity map and the vehicle‘s route.	
Shared Map	As local map approach, but the planned measurements are shared between all vehicles for an even better coverage.	

Table 11: Investigated probing strategies in the ICCOMQS framework.

the ICCOMQS framework. They are summarized and further illustrated by Table 11. We implemented these four strategies in our prototype of the ICCOMQS framework, which is based on Android clients and a Linux backend server, as described and used for our following works in Section 6.4.1. Furthermore we evaluated these strategies in a large scale simulation scenario as described in Section 6.3.4 to show the performance of the ICCOMQS framework in a scaled up environment to resemble a later real world deployment of the system. The traffic simulation is based on the SUMO traffic simulator used throughout our previous contributions of Chapter 4. The individual strategies are summarized in Table 11 and described in ascending order in correlation with their complexity as follows:

Trivial Approach

The most simple approach and our assumed baseline with which we want to compare our other proposed concepts is the Trivial approach. It executes one measurement of the wireless network, as soon as enough probing data (e.g. one megabyte) could be collected in the Scheduler unit. Thus the approach only relies on the amount of incoming data and requires near to no additional implementation effort. As it does not involve any further logic to improve the overall dispersion of the simulated test measurements, we assume it to perform as a lower boundary in the following Evaluation described in Section 6.3.4ff..

Aimed Random Approach

If we assume a scenario with a nearly constant rate of incoming data on each vehicle also the time interval for the next transmission of the Trivial approach is nearly constant. In consequence this effect might lead to a poor dispersion performance of the algorithm's designated measurement positions. To address this problem of the Trivial approach with only a bit more implementation effort we propose the Aimed Random approach. In contrast to the Trivial approach the Aimed Random approach randomly schedules the execution of the data transmission between the two time points when a new set of measurement data becomes available. Thus it shall achieve a better overall dispersion performance, with only slight implementation adaptations.

Local and Shared Measurement Map Approaches

The Local and the Shared Measurement Map approach are our true recommendations for well-performing probing algorithms to effectively built up and maintain the Connectivity Map. Therefore we extend our proposal to integrate the Connectivity Map as an additional layer into the HD Map, as described in the beginning of Section 6.2 and Figure 45. Besides the Connectivity Layer itself we also propose to integrate an additional layer into the HD Map, where the information regarding the already executed and planned measurements of the cellular network shall be stored.

This measurement planning layer is initially partitioned in equally spaced areas. The initial segment size is individually configurable and gets refined over time as explained in the following. The ICCOMQS framework schedules the execution of the measurements accordingly to the remaining segments of the map where no measurements have been already executed (white spaces) under consideration of the time constraints of the available user data. If no more white spaces are available or if the sensor data to be send cannot be delayed this far in time to reach the remaining white spaces another suitable grid with the lowest number of already executed measurements is selected. This segment then gets divided up into two smaller segments creating a smaller white space that is then scheduled to be filled with a new measurement's information. This process is repeated until a certain minimum segment size is reached. This minimum is configurable as well and depends on the assumed necessary spacing between each individual measurement to appropriately resemble the measured information (e.g. as in our chase the performance indicators for the cellular network) in the Connectivity Map. In our evaluation we assumed an initial segment size of 5km, which then gets reduced down to a minimum segment size of 10 meters between each individual measurement (see Table 13). The lower boundary of 10 meters was selected due to the measurement inaccuracy of currently available low-cost GNSS sensors (e.g. GPS or Galileo), which the vehicles use for their personal localization and thus also the planning of the measurements along their trip. This boundary value however might be further reduced in other scenarios if a more precise localization of the measurements is possible. Depending on the preferred degree of data protection and privacy, this general concept can be implemented in a Local or a Shared Map approach. Through the Local Measurement Map approach the measuring vehicle ensures its privacy, by only relying on the information about its personally executed measurements to plan the future ones. The Shared Measurement Map approach in contrasts shares all the planned and executed information between all vehicles and thus is assumed to achieve the highest degree of geographic dispersion as our considered performance metric for an effectively operating Connectivity Map.

Time Dependancy and Degradation over Time

As stated in the introduction of Chapter 6 and illustrated in Figure 2, not only the spatial refinement of our connectivity layer has to be considered. Through the several mentioned influencing factors also temporal changes have an effect on the overall achieved network quality. A very good example in our considered scenario therefore are the varying traffic patterns over the time, with dense traffic in the rush hours and only spare traffic in the off-peak hours. Another example are the changes introduced due to the network extensions of the cellular providers. Consequently older test-results have to be degraded by a time factor, to ensure the actuality of the Connectivity Map's information. In our opinion huge changes in the achieved measurement results therefore should be weighted stronger and trigger more follow-up measurements to quickly update the connectivity layer in such areas. The exact impact of these time influence factors however might be heavily dependent on the investigated scenario (e.g. as shown in Section 6.1 for the different areas considered in the Ko-HAF project) and consequently should be based on an extended measurement campaign to cover possible long term effects. Thus we consider the inclusion of such time patterns as future work to further improve the performance capabilities of the currently existing ICCOMQS framework. In the current status of the framework we focused on the optimization of the geographical coverage of the Connectivity Map as shown in the following evaluation. We argue that a sophisticated geographic dispersion of the network measurements will provide a larger benefit to the users of the Connectivity Map, due to a larger spatial variance of the cellular network measurements compared to their temporal variance, as presented in the Related Work of Yao et al. [195].

To evaluate the performance of the proposed probing strategies of the ICCOMQS framework in a large scale deployment we conducted a SUMO simulation similar to the ones described in Section 4.2.3 for the evaluation of our proposed Dynamic Map Update protocol and its HD-Wmap extension.

ratio of sensing vehicles in all road users	1/1000
simulated consecutive working days	20
required data for measurement [byte]	1000000
sensor data rate [byte/s]	5000
sensor data variance [byte/s]	±250

Table 12: Default simulation parameters for commuting scenario [93]

6.3.4 *Simulation Scenario and Configuration*

Cologne simulation scenario

We selected the TAPASCologne SUMO dataset for our simulation (see Figure 90 in the Appendix for details). It represents the traffic in and around the German city of Cologne. The scenario was selected, because it is known currently as the biggest traffic scenario available for SUMO as of the writing of this work (2019). Consequently it represents our requested large scale deployment excellently. The following evaluation results are based upon the dataset of the morning rush hour between six and eight o'clock with a peak of 40.000 commuting vehicles around 7 o'clock. The simulation was executed in twenty consecutive runs, to resemble the pattern of commuting traffic during the work days in and around Cologne for a whole month. We simulated for this long period of time to reach possible saturation effects of the geographical dispersion achieved by the different probing strategies, which would then require further probing strategies focusing on the time degradation of the data. The detailed settings used during the simulation are summarized in Table 12. From the full set of vehicles we assumed 1000 to be capable to share their sensor data with the ICCOMQS framework. Further we specified the amount of one megabyte of data necessary to be collected to execute a measurement in resemblance of our conducted measurements. Furthermore we assumed the sensor data collecting vehicles to achieve a data rate of 5000 byte/s. We achieved similar rates with our own prototypical implementation of the ICCOMQS framework, which we used throughout our subsequent contributions to collect the thereby required sensor data (see Section 6.4.1 for details). The achieved data rate values however might change over time, depending on the type of collected sensor data and experienced environmental changes. An example therefore are the different amounts of lane markings that are detected in different traffic environments, e.g. highway or city streets. Thus we further implemented a variance of ±250 byte/s in our generated data stream to resemble this influencing factor in our simulation as well.

6.3.5 *Initial Graphical Evaluation*

As our first evaluation step we performed an initial graphical comparison between the different proposed probing strategies to identify their achieved coverage of geographic area, as our introduced performance metric. Therefore we calculated heat maps of the complete scenario area for each proposed strategy, shown in Figure 50.

Heat maps for performance visualization

The colorization of each heat map is scaled individually to match with the related approach's dispersion density of measurements. Red segments identify areas with a high density of executed measurements, whereas yellow sections represent areas with a lower density of measurements. About 5% of all executed measurements are located in the light yellow areas, with an increasing number towards the dark red zone. Larger areas of the same color correlate with a more equally dispersion of the measurements, thus they indicate a better performing probing strategy. For the creation of the images and the required initial calculation steps we relied on the free Geographic Information System QGIS[26]. Figure 50 shows the achieved results.

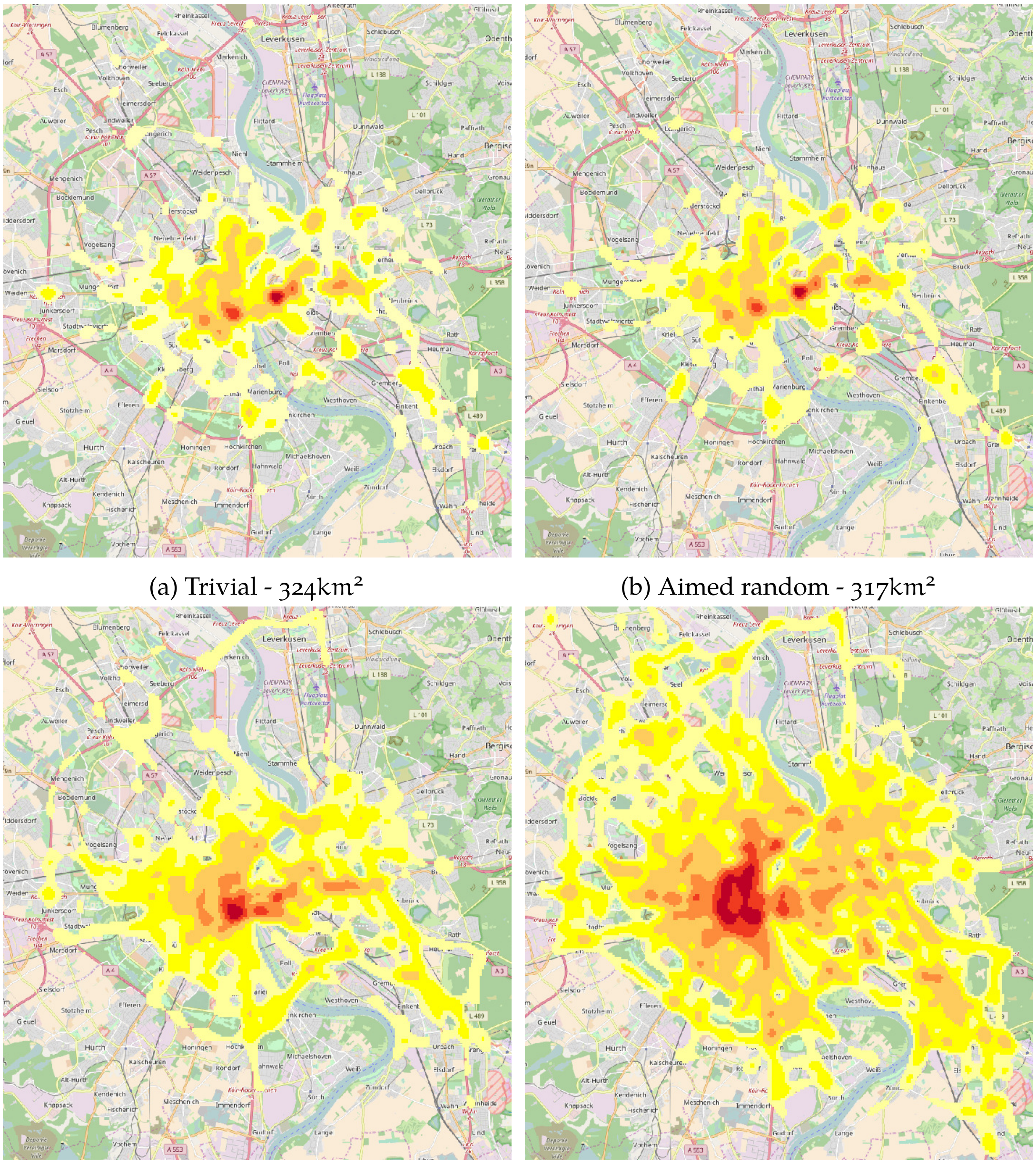
The Trivial approach as illustrated in Figure 50a achieved a total coverage area of about 324 km^2 after the simulated period of 20 consecutive days with the same pattern of commuting traffic (see the Graphs in Figure 52 for further reference). In total our considered set of 1000 vehicles generated probing data for 67588 cellular measurements. The Trivial approach used all of the data as it sends it out right away. The other three proposed probing strategies conducted less measurements throughout the simulation, as they scheduled the transmission of data in comparison to the Trivial approach. Less probing data to cover the same area of the map is beneficial in a way that the remaining data can be used for either other areas or for the intelligent maintenance of the Connectivity Map over time. The Aimed Random approach as visualized in Figure 50b conducted 60298 measurements in the same time and covered an area of around 317km^2. Consequently the approach could cover nearly the same geographic area (only 2% less) as the Trivial approach, by using 11% less data. In contrast to the aforementioned concepts the Local Map and the Shared Map approach performed significantly better. The Local Map probing strategy shown in Figure 50c could cover an area of 548 km^2 with only 51644 conducted measurements. This is an increase of 69% in covered area and a reduction of 24% in probing data required therefore compared to the Trivial approach. As expected the Shared Map approach visualized in Figure 50d achieved the best coverage performance of 801 km^2 with 47290 conducted measurements. This is an increase in the covered geographic area of 147% compared to the Trivial approach, with a reduction in the required measurement data of about 30% respectively.

Besides the shown visualizations of the complete scenario area we furthermore performed an in-depth graphical evaluation between the different approaches. Therefore we selected a small, designated area in the city center near the Cologne Cathedral and the main station as indicated by Figure 51.

The four individual graphics show the dispersion of measurements for each proposed strategy. Figure 51a clearly visualizes that the Trivial approach conducted most of its measurements on the busy main streets. Large areas of less frequented side streets where nearly skipped. Thus the Trivial approach leaves out important connectivity information of those areas. The Aimed Random (Figure 51b) and the Local Map approach (Figure 51c) achieve a similar improved performance compared to the Trivial Approach. Both approaches include several additional side streets into their coverage.

---

[26] http://www.qgis.org/en/site/ (Last accessed on August 1, 2019)


(a) Trivial - 324km²

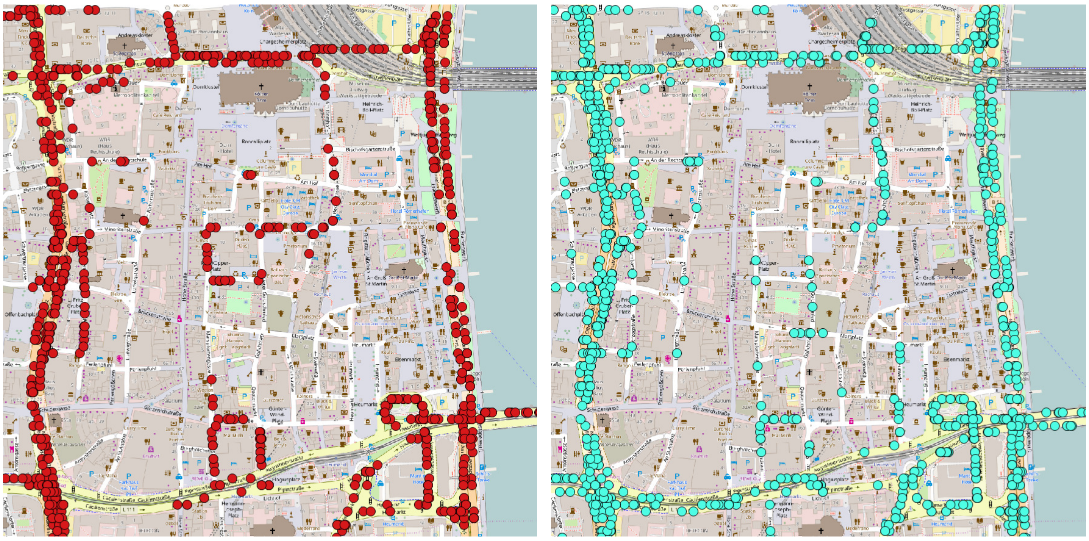
(b) Aimed random - 317km²

(c) Local map - 548km²

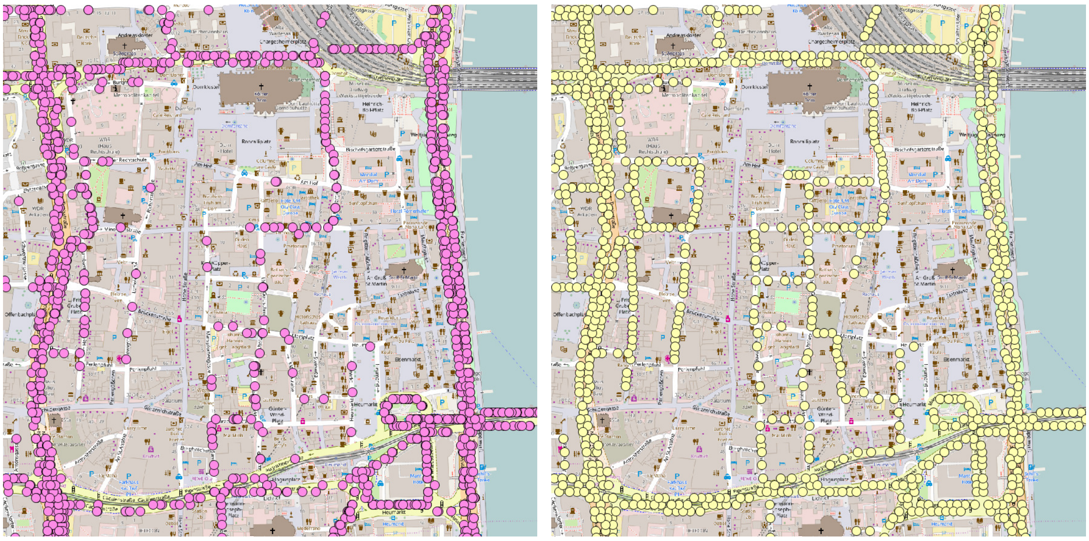
(d) Shared map - 801km²

Figure 50: Heatmap of the dispersion of 95% of the simulated measurements after 20 consecutive days of commuting [93]. Map data ©OpenStreetMap contributors

The Shared Map approach (Figure 51d) furthermore outperforms these two concepts, with a very evenly distributed field of measurements over all covered streets.



(a) Trivial (b) Aimed random



(c) Local map (d) Shared map

Figure 51: Detail comparison of the different measurement dispersion approaches in the inner city area of Cologne near the main station [93]. Map data ©OpenStreetMap contributors

6.3.6 *Statistical comparison*

In our second evaluation step, to compare the different dispersion strategies on a reliable numerical basis, we calculated the Nearest Neighbour Index (NNI[27]) as perfor-

---

[27] http://www.geoib.com/nearest-neighbor-index.html (Last accessed on August 1, 2019)

	Parameters		Day				
Approach	Data rate variance [byte/s]	Measuring distance [m]	1	5	10	15	20
Trivial	±250	-	0.457	0.026	0.014	0.009	0.007
	±2500	-	0.454	0.157	0.120	0.099	0.084
Aimed random	±250	-	0.450	0.279	0.229	0.202	0.181
Local map	±250	5000 - 10	0.530	0.353	0.297	0.265	0.244
Shared map	±250	5000 - 10	0.796	0.528	0.435	0.388	0.360

Table 13: NNI over twenty days for examined approaches [93]

mance metric. The NNI describes the spatial dispersion of the executed measurement points in a defined geographic area. It is calculated as described by Formula 7.

$$\begin{aligned} \mathrm{NNI} &= 2 * \overline{D} * \sqrt{N/A} \\ \overline{D} = \text{average distance between each point and its nearest neighbour} &= \textstyle\sum d/N \\ d &= \text{each points individual distance to each other} \\ N &= \text{number of studied points} \\ A &= \text{size of studied area} \end{aligned} \tag{7}$$

The NNI thereby spans between the values of 0 and 2.15. In our chase 0 is considered as the worst achievable value that one of our dispersion concepts could reach, as it represents one single clustered area of data points, whereas 2.15 is equal to a regularly dispersed pattern of points, the best case in consideration of a beneficial, large geographic coverage. An NNI value of 1.00 indicates a random dispersion of all considered points.

Higher NNI value is better

For our evaluation we calculated the NNI for each of the probing algorithms after every period of five consecutive days (resembling a full working week) using QGIS. QGIS directly calculates the NNI values from a provided geographical dispersion of the vehicles, which we obtained through our SUMO simulation. The obtained NNI results are presented in Table 13 and Figure 52.

After the first simulation day the Trivial and the Aimed Random probing approach achieved comparable NNI values of 0.457 for the Trivial and 0.450 for the Aimed Random approach. The Local Map and the Shared Map approach, as expected, performed significantly better with achieved values of 0.530 (Local Map) and 0.796 (Shared Map). As of the first day no previous measurements have been conducted and thus all four

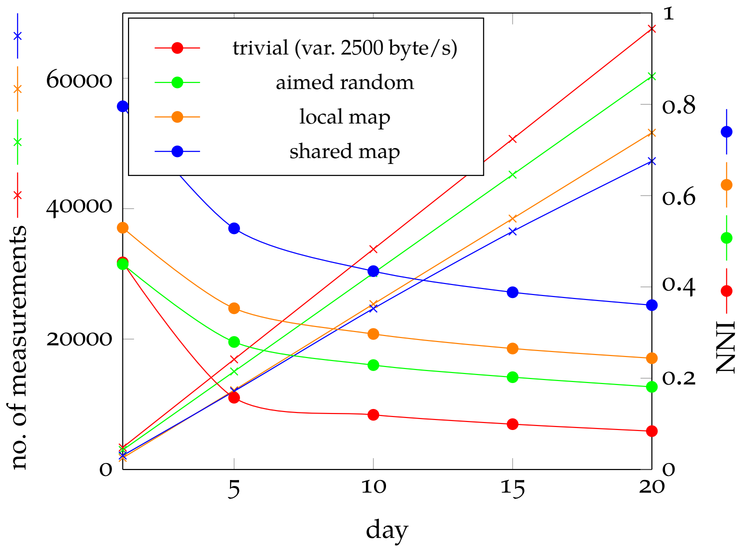


Figure 52: Plot comparing scheduling approaches considering number of measurements and NNI over twenty days [93].

approaches achieve comparable high values of the NNI with slightly improved performance results of the map based probing concepts over the other two suggested algorithms. In the following days the NNI decreases continuously, as more and more traveled roads get covered more densely with an increasing number of cellular measurement points. This rate becomes less and less over time as the new incoming measurements have less impact on the overall density of the dispersion of the set of already conducted measurements, as visualized through the gradients of the curves in Figure 52. As initially assumed the Aimed Random approach always achieved a better performance than the Trivial base line concept. We consider the low variance of only ±250 byte/s in our simulated data stream as one major negative influencing factor on the Trivial probing. The low variance lets the Trivial approach schedule its measurements very closely near to each other every day of the car's commuting trip. We further investigated this aspect by increasing the data variance received by the trivial approach up to ±2500 byte/s. As result the Trivial approach's performance increased. However it still remained below the one of the Aimed Random approach with less variance in its data stream, as plotted in Figure 52. This justifies our initial assumption that the Aimed Random approach outperforms the Trivial approach by only requiring a slight additional implementation effort. The two map based approaches (Local and Shared) in comparison outperformed the Aimed Random approach over the full twenty days of simulation. At the end of the simulation the Shared Map concept achieved an NNI nearly twice as high compared to the one of the Trivial approach, as expected the best overall performance result.

To summarize our evaluation results a decision for one of the probing strategies has to be based not only on the achieved performance results, but also on the defined side conditions. Overall the Shared Map approach achieved the best performance results. However if a reduction in the communication overhead is desired or privacy concerns are present the Local Map approach is the most suitable one. If furthermore computation capabilities or storage space on the client side are an important criteria the Aimed Random approach is a suitable choice. It outperforms the Trivial probing concept, with nearly no implementation overhead.

ICCOMQS prototype

Besides our extensive simulations we furthermore implemented an actual working prototype of the ICCOMQS framework as a combination between an Android based smartphone client and a linux based backend server as discussed in detail in the following Section 6.4.1. This prototype was used as a fundamental prerequisite for our subsequent contributions.

In summary the ICCOMQS framework enhances the performance of a Connectivity Map in the anticipation of geographic variances in the quality of the cellular network. Therefore the framework efficiently and intelligently conducts cellular network measurements. The successfully maintained Connectivity Map itself then improves the overall data transmission performance of the self-driving vehicles.

From the achieved promising evaluation results of this work, we now shift the focus of our research upon the anticipation of temporal influencing factors on the performance of the cellular network (e.g. weather effects or the maintenance of a cell tower) in the following Section 6.4.

6.4 CONNECTIVITY MAP SUPPORTED ONLINE THROUGHPUT ESTIMATION

As expressed in Section 3.3.3 of the Related Work, the concepts currently at use to estimate the cellular network performance under temporal influences most commonly rely on machine learning techniques. Therefore the required network performance indicators are directly collected (online) from the communication devices, which are installed in the vehicle. This data is then fed as input into a trained machine learning algorithm. That way the various proposed algorithms predict the nearby networks future quality for some seconds till a few minutes ahead on the trip. We focus our contributions on the estimation of the achievable throughput as we consider it the most important criteria to ensure a reliable functionality of the HD Map. The major disadvantages of the investigated Related Works are summarized as follows:

Focus on throughput prediction for HD Map maintenance

i) Many of the Related Works [68–70, 73, 74] rely upon specialized measuring hardware or software tools (e.g. QUALCOMM's eXtensible Diagnostic Monitor) to obtain their training and testing data for the used machine learning algorithm. This furthermore often includes provider internal information, which is otherwise not publicly available. As result the mentioned aspects especially hinder a possible future large-scale deployment of the proposed concepts as considered by us for the context of self-driving vehicles.

ii) When training their machine learning algorithms the proposed approaches consider the amount of collected network performance measurements, which correlate with the achievable throughput, as one single set [64, 66–72, 74, 155]. Many of the Related Works do not even consider a mobile application scenario in their evaluation.

Based on our contributions in the context of Connectivity Maps, we consider this global training set as a major drawback in the estimation process, as it likely obfuscates the spatial features of the environment and the deployed infrastructure in the training process.

Cell-specific training

To overcome these disadvantages we performed scientific contributions as follows: We proposed and verified upon real world data a concept to subdivide the gathered training data into cell-specific training sets (Sec. 6.4.1). That way the machine learning algorithms are able to include the spatial features of the cellular network in the learning process on a per cell tower level (see for example Fig. 54a) similar to the working principle of a Connectivity Map. Thus we achieve a much higher prediction accuracy compared to the mentioned global training data set, which is common for the related online estimation concepts. To compare our proposed locally trained online estimator with the common globally trained estimators of the Related Work, we chose the Random Forest regression algorithm as used by Samba et al. [69] as our reference algorithm. As one of the most recent Related Works Samba et al. achieve highly accurate estimation results by using this algorithm.

6.4.1 *Android Measurement Application and MobileInsight Extension*

Release of own benchmark data set

For the in depth investigation of our proposed online throughput estimator, based on localized training data, we further extended the capabilities of our custom Android application, which we developed during the work upon the ICCOMQS framework (see Sec. 6.3 and Fig. 53). That way we were able to rely upon common smartphones as our measuring probes, instead of the hardware used during our previous measuring campaign (Sec. 6.1.4). By relying upon such cost-efficient devices, which provide a similar set of various network quality indicators compared to the built-in communication hardware (see Table 15), we support the future research in this domain. As we make our obtained network measurements freely available on GitHub[28] we provide a valuable public benchmark set to the research community. That way future works can easily compare their own achieved performance results on a common and affordable hardware basis. Furthermore by deploying our measuring application on several smartphones of the same kind we were able to measure the three cellular providers in parallel (in the following identified as providers A, B and C), which maintained LTE networks in Germany, during our test drives (May 2018). That way we could exclude additional influencing factors from our collected data set such as variations in the device internal hardware, as well as the time or the location when a measurement was conducted.

Based on the ICCOMQS prototype, our Android application, illustrated in Figure 53, is able to periodically measure the throughput of the current cellular network in the upload and download direction. Therefore the application sends sensor data from the smartphone to a central server and vice versa by using the UDP protocol. The server

---

[28] https://github.com/florianjomrich/cellularLTEMeasurementsHighwayA60 (Last accessed on August 1, 2019)

itself was hosted at the Technical University of Darmstadt, to ensure a sufficiently high network capacity on the server's side, making the cellular network the bottleneck of the end-to-end channel. By using the UDP protocol we ensure that the measurement of the cellular network is not somehow influenced by protocol induced behavior, such as slow start or congestion control as it is the case when using the standard configuration of the TCP protocol. The collected throughput estimations and their measured correlating network performance indicators are then stored in a PostgreSQL database for further processing towards their usage as training or testing data in our online estimator. To ensure a reproducible probing of the cellular network at its capacity limit we enhanced the existing upload and download sending process by relying upon the cellular networks performance indicators (e.g. the available transmission bandwidth) obtained from the smartphone itself as described in the following (see Sec. 6.4.1 and Sec. A.12 in the Appendix). With this information at hand in correlation with the detection of packet loss during the transmission process we ensured to always utilize the cellular network's full available capacity by adapting (increasing or decreasing) the transmitted payload size accordingly. This for example is beneficial after a handover between two different cells (with respectively different capacities) has been executed. Based on the lower layer information our application can adapt itself much quicker to probe the new cell tower at its capacity limit, compared to for example the TCP protocol that can only rely on the detection of packet loss to adapt its transmission process. This consequently saves probing data, which is especially useful in consideration of our vehicular use case, where such cell changes frequently happen.

Lower layer information used for adaptation of measuring

As stated previously a major concern during the design of our online estimator and the implementation phase of the Android-based measuring application was an easily possible future large-scale deployment of our proposed concept (in contrast to the Related Works). Consequently our first proposed mechanism (1.)) as visualized in Figure 53 works "out-of-the-box" simply by installing the application on the considered smartphone. In consequence this first approach obtains its complete set of network quality indicators (Tab. 15a) used for the later online throughput estimation via the publicly provided APIs of the Android operating system.

Two different approaches proposed and investigated

To investigate the impact of additional cellular network performance indicators on the achievable throughput estimation performances, which were not obtainable from the Android API during our test drives (May 2018) we developed our second mechanism (2.)) visualized in Figure 53. This set of features for example included the availability of Carrier Aggregation or MIMO along the selected highway segment. For this required extension however we did not want to spend further money on additional hardware or software licenses, as for example required for the utilization of QUALCOMM's eXtensible Monitor Software (QXDM), which is frequently used throughout the Related Work. By investigating further Related Work we came across the so called MobileInsight Android application developed by Li et al. [196]. MobileInsight is a free and open source application, that provides nearly the same network quality features as the QXDM toolkit. MobileInsight achieves this by decoding the provided information of the same interface on the LTE modems chipset (the QUALCOMM debug port). Our measuring application then received these decoded network quality parameters via

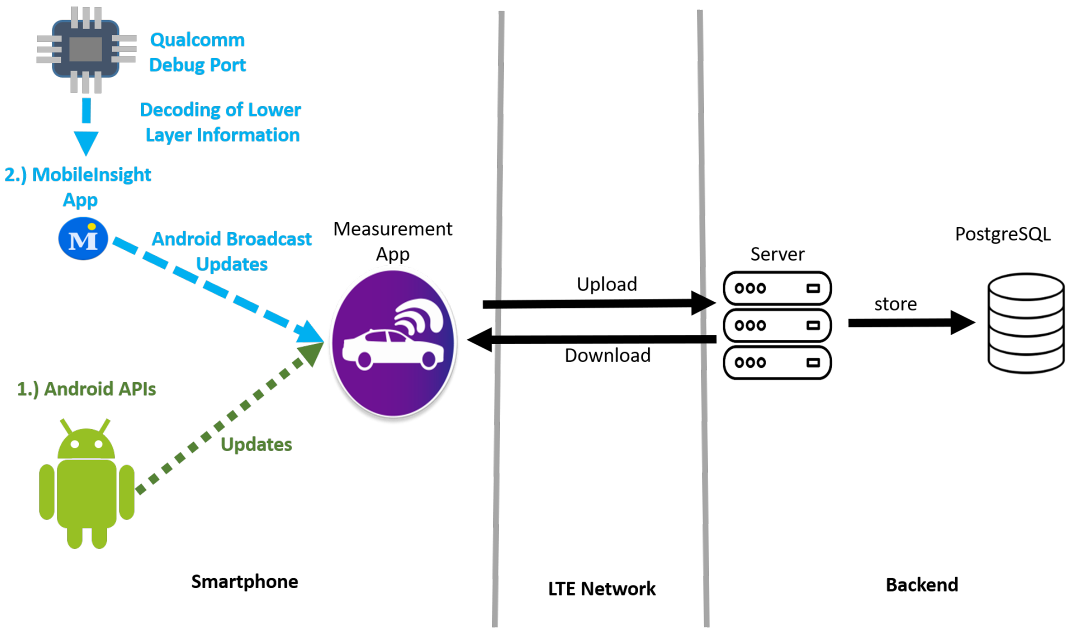


Figure 53: Extension of the ICCOMQS frameworks prototype application to obtain cellular network performance indicator information via two different sources [95]. Namely: 1.) The performance indicators are obtained through the public and open Android API (Android 7.1.2). 2.) The performance indicators are obtained via Android Broadcast messages from the MobileInsight application developed by Li et al. [196]. The MobileInsight application itself decodes the information related to the end devices experienced current network quality directly from the device's LTE modem using the chipset's debug port interface.

Android Broadcast messages. The extended list of network performance indicators, which could be collected by us this way are stated in the Tables 15b and 15c. The only disadvantage of this second approach compared to our first proposition is the fact, that MobileInsight requires root access on the smartphone to operate. However compared to the investigated Related Works (e.g. [68–70, 73, 74]), which relied either on special hardware or costly licensed software to obtain the same network features for the prediction process, we consider our approach a rather easily deployable and cost efficient way to obtain the required performance indicators. With the further development of the mobile operating systems (e.g. Android and iOS) also more and more features naturally get included directly into the platforms and do not require specialized software such as MoblieInsight. The indication of the availability of Carrier Aggregation is such an example. During our evaluation, as explained in Section 6.4.1, we relied upon Nexus 5X smartphones running MobileInsight on an Android 7.1.2 platform. With the introduction of Android 8.1 the availability of Carrier Aggregation is also now indicated through one of the public APIs of the operating system.

A full list of all the parameters related to the cellular network's quality, which our custom Android application was able to obtain through the two different interfaces, is presented in Section A.12. Not all of those parameters were directly used during the machine learning process, but were necessary to create the initial training data set for the global and the localized training approach as described in the following Section 6.4.1.

Highway Evaluation Scenario

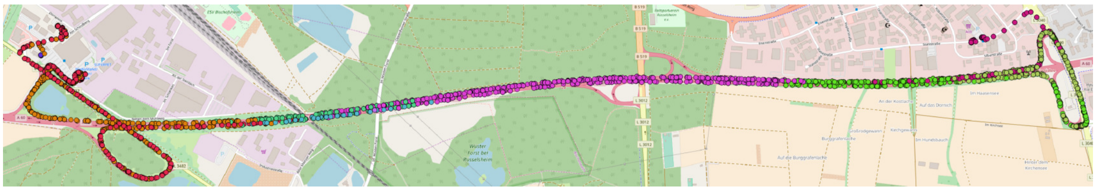
In opposition to many of the Related Works, we focused our own measuring campaign especially upon a mobile scenario. Namely a highway scenario, as highly automated driving functionalities most likely will become first available in such traffic environments (see for example the test area in Section 6.1.1).

To be able to obtain a sufficient amount of labeled throughput measurements in a reasonable amount of time to train and test our proposed machine learning concept, we selected a section of the highway A60 nearby Rüsselsheim as our test site (Fig. 54). In total we conducted two different measuring campaigns, one for each of the previously explained approaches (Sec. 6.4.1). For the collection of the throughput estimations we exclusively focused on the LTE network, which was available along the highway as the most advanced cellular network. We ensured our collection process by forcing the modems of the used smartphones in an LTE exclusive mode. Hence no other network technology was used.

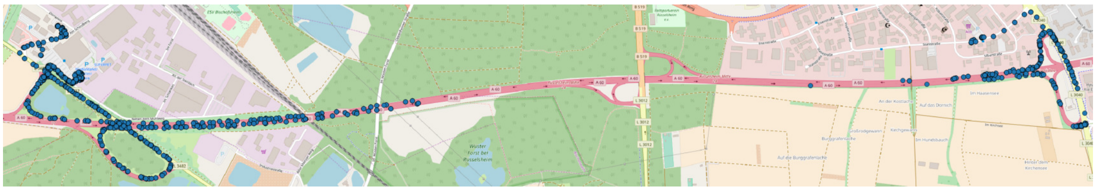
Two measuring campaigns

The first campaign focused on the evaluation of the proposed estimation approach, that relied upon the Android API as source of the network quality indicators. During this first campaign of ten days of test drives, we were able to obtain over 74.000 sets of throughput estimations and correlated network quality parameters (for further details see [94]). Based on the obtained results of our first campaign we rethought our estimation concept and developed our second proposed mechanism. Afterwards we conducted a second measurement campaign over a period of three consecutive days. In this campaign we compared our first approach (Android API), with our second proposed mechanism. This second mechanism is based upon the network quality information directly obtained from the smartphone's LTE modem using the MobileInsight application [196]. To compare the performances of both approaches we collected the network quality parameters provided by the Android API (Tab. 15a) in parallel to the data provided by the MobileInsight application (Tab. 15b and 15c), while executing our test measurements. All measurements of the second campaign have been conducted using Google Nexus 5X smartphones (running on Android 7.1.2). We used the Nexus 5X smartphone as our device of choice, as it is capable of using 2-way Carrier Aggregation and 2x2 MIMO, advanced LTE features, which were available on our highway test section. We consider those features as described in Section A.3, highly important for our proposed local training concept, as they are available only at certain areas along the highway test segment. Figure 54b illustrates this for the availability of Carrier Aggregation for provider A along the track.

During this second campaign we obtained over 45.000 throughput estimations via the Android API and over 540.000 estimations from the MobileInsight application for the training of our Random Forest throughput estimator. See Table 14 for an in depth view on the data set regarding each of the three providers. The different amounts of collected throughput estimations result from the varying sampling interval provided through the Android API and the MobileInsight application (see Figure 55).



(a) Measurements colorized accordingly to the serving cell towers unique ID (Cell ID), that was used for the data transmission.



(b) Measurements with available Carrier Aggregation of Provider A

Figure 54: Visualisation of the 5km long Section of the German highway A60 near Rüsselsheim where our measurement drives have been perfomed [95]. Map data ©OpenStreetMap contributors

As the second campaign included the first investigated approach as well in its evaluation, we will focus on its results exclusively in the following Section 6.4.2. For the similar evaluation results obtained in the first campaign see [94].

Provider	A	B	C
Download	8369	8571	6027
Upload	8367	8566	6002
Overall	16736	17137	12029

(a) Collected Android Data Points

Provider	A	B	C
Download	100920	105377	50506
Upload	149166	89847	46653
Overall	250086	195224	97159

(b) Collected MobileInsight Data Points

Table 14: Amount of throughput estimations with correlating network performance indicators collected during the second measuring campaign [95].

Used Features for Machine Learning

As common initial step in the machine learning process the measured data set has to be split into a training set to train our selected Random Forest estimator and a testing set to verify its learned estimation capabilities in the subsequent evaluation (Sec. 6.4.2). As common for a supervised machine learning task we divide our collected data set up into 70 % training and 30 % testing data. The collected network quality parameters, which we obtain via the Android API and the MobileInsight application, which are used as features for the machine learning estimation process are summarized in Table 15.

RSRP, RSSI, RSRQ, Longitude, Latitude, Speed of the Vehicle, Timing Advance

(a) Machine learning features provided by the Android APIs for Upload and Download direction

RSRP, RSSI, RSRQ, Longitude, Latitude, Speed of the Vehicle CQI, Carrier Aggregation Availability, Modulation and Coding Scheme, MIMO

(b) Machine learning features provided by MobileInsight for the Download direction

RSRP, RSSI, RSRQ, Longitude, Latitude, Speed of the vehicle, CQI Modulation and Coding Scheme

(c) Machine learning features provided by MobileInsight for the Upload direction

Table 15: Machine learning features provided by the Android approach (a) and the extended MobileInsight approach (b, c) using Nexus 5X smartphones [95].

When using the Android API as our choice of data source, we can concatenate each of our active throughput measurements of the cellular network with one correlating set of network quality indicators, which are passively measured during the same time interval. This is possible due to the rather low sampling rate of the Android API (about one second per set of indicators) in correlation with a fast completion of the full data transmission (less than one second). This exact one on one mapping however is not possible, when considering the network quality data provided by the MobileInsight application due to its increased sampling rate of only a few tens of milliseconds between each set of network quality values. Figure 55 exemplary showcases the sampled values of the Reference Signal Received Power (RSRP) over a measuring period of 11 seconds. During this time the MobileInsight application provides about 281 samples for the RSRP value, whereas the Android API only achieves 6 sampled values in the same amount of time. Furthermore Figure 55 showcases the high timely variance in the cellular networks quality, which is exposed much better through the MobileInsight interface in comparison to the Android API. We assumed a huge potential in this higher time accuracy to improve the prediction capabilities of our throughput estimator. To maintain this time accuracy, we had to find a different solution, as a mapping of each obtained MobileInsight value onto our latest achieved throughput measurement result is incorrect and obfuscates the timely performance fluctuation in the cellular network. Consequently to achieve a correlating higher time resolution regarding the current throughput of the connection, we rely upon the value of the transport block size, which the serving cell tower allocates during the measuring interval for the client device. This additional value is provided through the MobileInsight interface as well. Thereby, the transport block size is derived from two further values (as described in detail in Section A.3): i.) the amount of allocated resource blocks out of the set of all

High time accuracy of MobileInsight compared to Android API

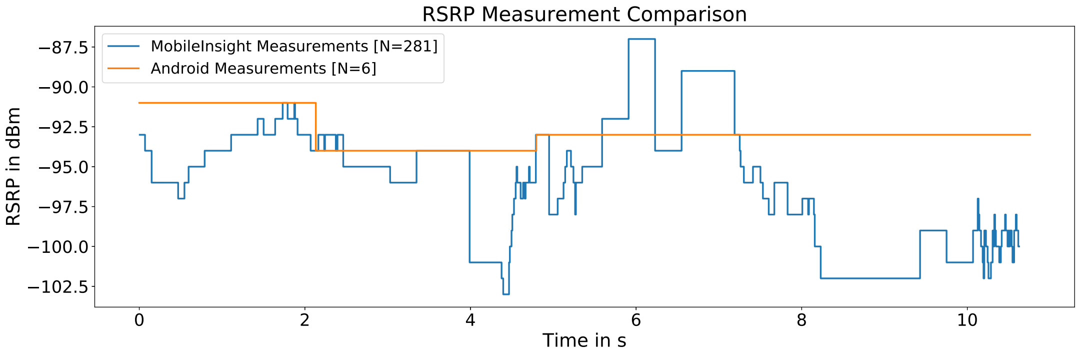


Figure 55: Comparison between the measurement accuracy of the Reference Signal Received Power (RSRP) between the stock Android API and MobileInsight. Clearly visible is the highly increased timely resolution of MobileInsight in contrast to the Android API [95].

available resource blocks, which the serving cell tower assigns every millisecond anew between all its clients and ii.) the current modulation and coding scheme, which the client uses based on its currently received channel quality (a stronger signal allows a higher modulation and coding scheme). By dividing the allocated transport block size through the MobileInsight measurement interval of some milliseconds, we are able to calculate a throughput estimation of the current connection for this time interval. This process however is only executed to obtain a labeled training data set for our machine learning process. We do not include the amount of allocated resource blocks or the resulting transport block size as features in our training data set as they are directly correlated with the throughput value. Furthermore, to obtain these values requires an active data transmission to be already present during the estimation process (as for example considered by Yue et al. [155]). For our investigated context of self-driving vehicles, the estimation algorithm instead shall enable the instantaneous decision process, whether or not data should be transmitted at the currently reached position along the track, without wasting any data during the process itself. These same prerequisites regarding an instantaneous throughput prediction were considered by Samba et al. [69] as our chosen reference work.

6.4.2 *Performance Comparison*

To compare ourselves with the achieved performance of the estimator generated in the work of Samba et al. [69] we rely upon the same performance metric: the R^2 value as described by formula 8. In the formula $\bar{y}$ is the average of all considered throughput measurements, y_i is the current throughput estimation and $\hat{y}_i$ is the predicted throughput value. Consequently, a higher R^2 value indicates a better estimation performance. An R^2 value of one would resemble a perfect estimation, where each estimation $\hat{y}_i$ is exactly the currently measured value y_i.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{n}(\bar{y} - y_i)^2} \tag{8}$$

In the following, we exemplary showcase the achieved evaluation results for provider A. However the results obtained for provider B and C are similar as shown in Section A.13 in the Appendix. Consequently, we are able to generalize our statements described in the following for all three cellular providers.

As a first evaluation step we investigated the overall distribution of the conducted throughput measurements in Section 6.4.2. Therefore we compared the measured results using our Android application with the obtained throughput values provided directly by the LTE chipset via MobileInsight. The results for provider A are presented in Figure 56.

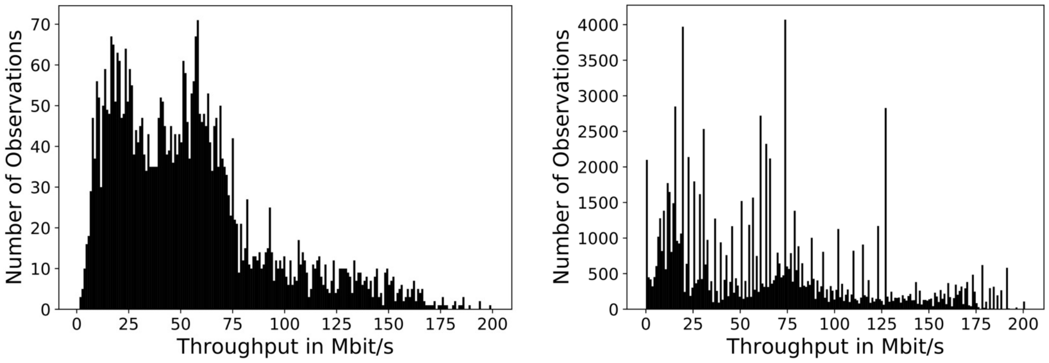
Throughput Histograms

By comparing the histograms of the upload and download data rates measured by the Android Application as shown in Fig. 56a and 56c with the values of the Mobile Insight API in Fig. 56b and 56d the diverse distribution of the values is clearly visible. The histograms of the Android measurements show a drop in the distribution of the achieved throughput values at an upload speed of about 25 Mbit/s and a download speed of around 75 Mbit/s respectively. The distribution of the throughput values provided by the MobileInsight interface in comparison is even more diverse. This is likely correlated to the higher time accuracy provided by MobileInsight as discussed in Section 6.4.1 and illustrated by Figure 55. The throughput values of the MobileInsight application are derived from the allocated resources in the investigated LTE network. These resources are quickly reassigned by the providing cell tower between all its active clients after each period of milliseconds.

Android and MobileInsight Performance Comparison

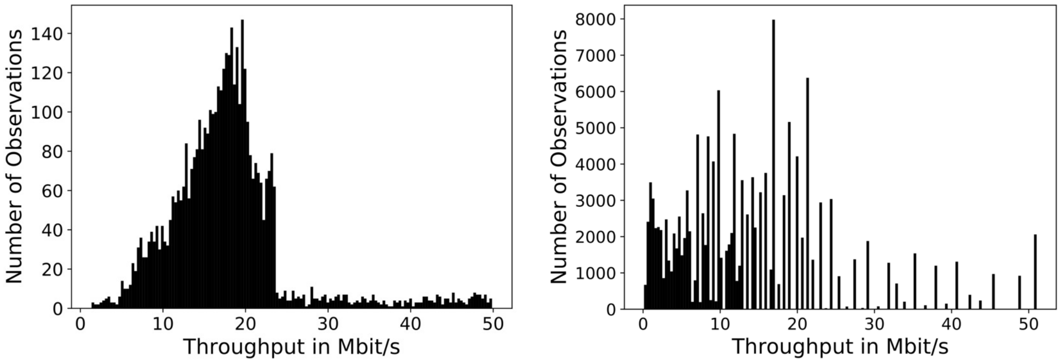
In the following evaluation step, we compare the prediction performance of our chosen Random Forest estimator, when being trained with the Android data set compared to the MobileInsight data set. To be comparable to the performance results obtained during the Related Work of Samba et al., the training process therefore is conducted in the same way with the full set of all collected measurements available as global training data set. The achieved results are shown in Figure 57.

The Android API based estimator, with its comparable low time resolution of collected network performance indicators, achieves positive R^2 performance results peaking at about 0.4. These evaluation results clearly indicate, that a practical estimation of the throughput value is possible even when relying only on publicly available interfaces, which work "out-of-the-box" and will only be improved in the future through continuous development and the addition of further quality parameters.



(a) Android Download Measurements - Provider A

(b) MobileInsight Download Resource Allocation / Measurements - Provider A



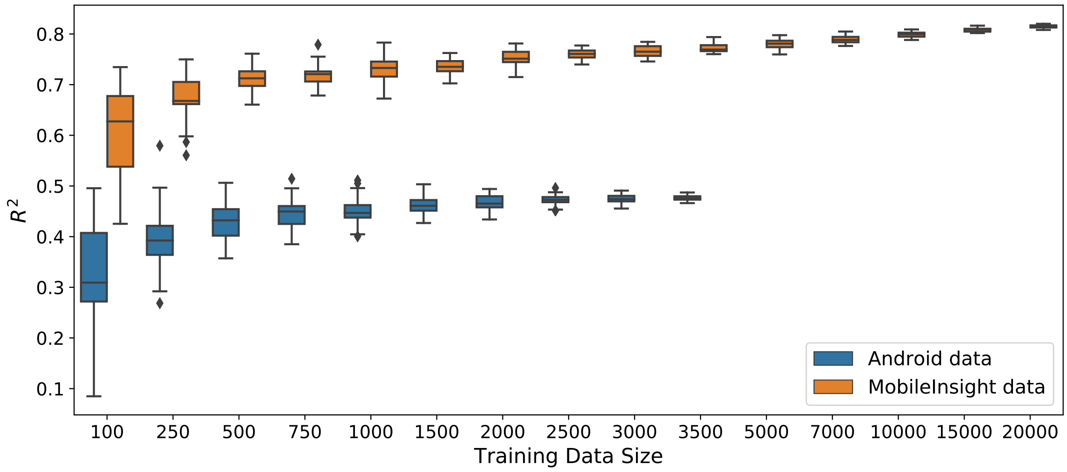
(c) Android Upload Measurements - Provider A

(d) MobileInsight Upload Resource Allocation / Measurements - Provider A

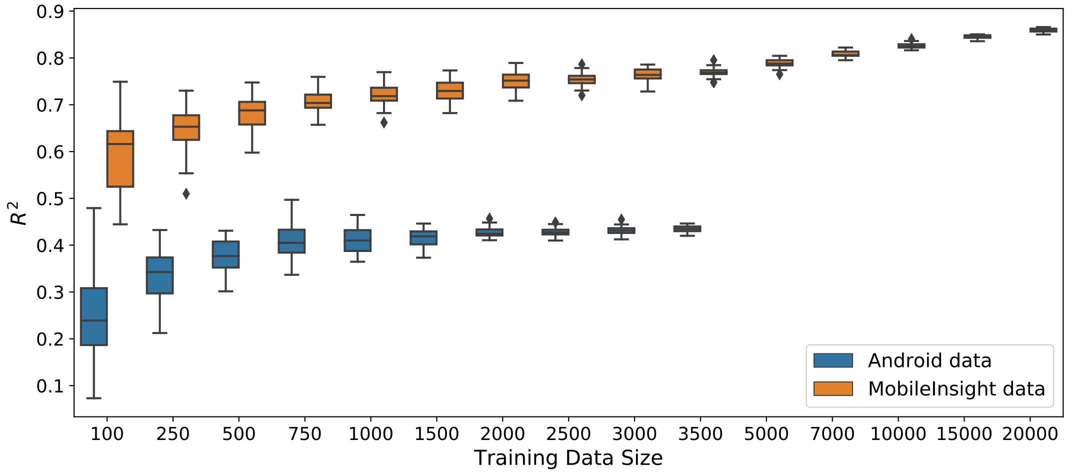
Figure 56: Histograms of the distribution of throughput measurements obtained from Android and MobileInsight [95].

Our estimator based on the data provided by MobileInsight in comparison achieves similar performance results for the R^2 metric as the estimator in the work of Samba et al. [69]. Depending on the transmission direction our throughput estimator achieves R^2 values of about 0.9 for the upload and 0.8 for the download direction. The highest R^2 value achieved by the estimator of Samba et al. for the download direction is 0.85 as indicated in Figure 6 in [69]. However to achieve these results, the authors rely upon features in the training data, which are only available through the back end of a cellular network provider. This was only possible through additional support in their evaluation as such information is not commonly available. Our approach achieves similar results, but instead only relies on features, which are directly obtained from the user end devices. In our opinion, this fact is a major advantage for a future large-scale deployment of our proposed estimation technology.

By comparing the two estimation approaches with each other, the comparably fast saturation in the prediction accuracy of the Android based approach becomes visible. The Android API based estimator saturates at a training data size between 2500 - 3500 samples, most likely due to its comparable coarse resolution in time. The MobileInsight based estimator instead benefits much more from an increase in the training data,



(a) Download throughput prediction performance comparison



(b) Upload throughput prediction performance comparison

Figure 57: Performance comparison between Android and MobileInsight [95].

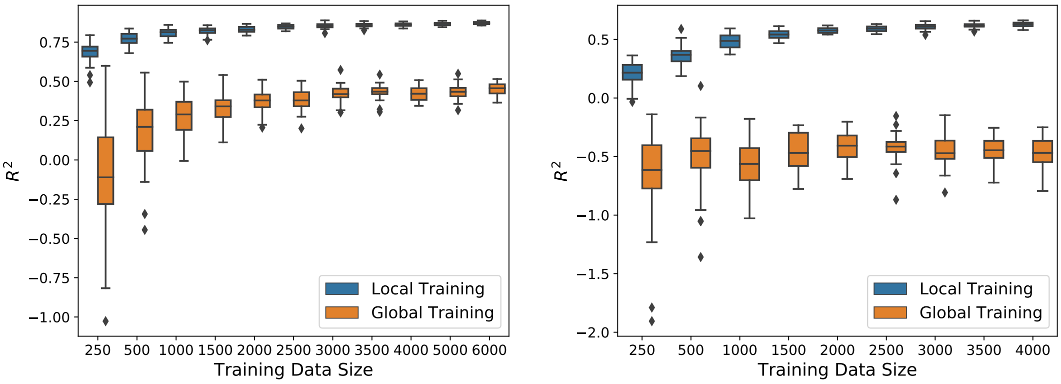
indicating a better prediction model. Even at an evaluated training data size of 20.000 samples, no clear saturation of the models performance became visible.

Localized vs Globalized Training Based on MobileInsight Data

To evaluate the main contribution of our work, the proposed localized training approach in comparison to the commonly used globalized training, as shown in the previous Section 6.4.2, we had to conduct further preparation steps. For each measured cell, we created a cell-specific local training set, which only contained the measurement samples obtained from the specific cell itself. The remaining measurements then formed the related global training set, which is not location-specific.

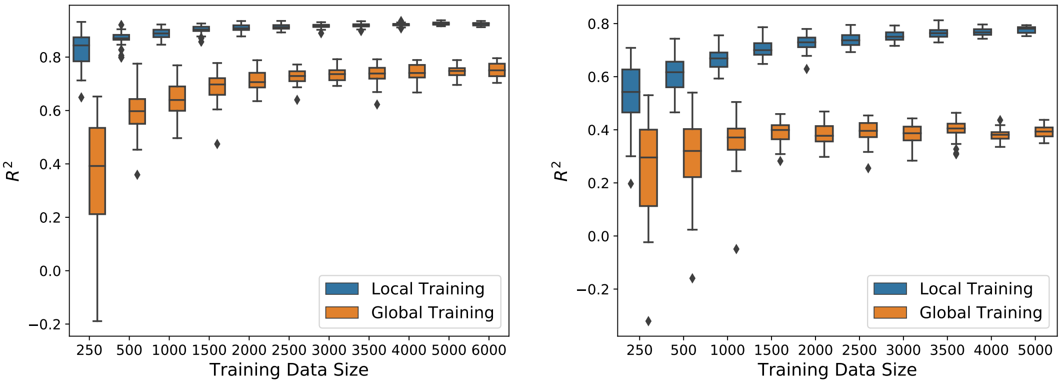
Localized training significantly outperforms common globalized training for all investigated cells

The obtained R^2 values for the estimation of the upload and download throughput of the four most frequently measured cells of the investigated provider A (identified as



(a) Downlink - Cell A

(b) Uplink - Cell A



(c) Downlink - Cell B

(d) Uplink - Cell B

Figure 58: Performance comparison between localized and global training data of the first and second most often measured cells of provider A [95].

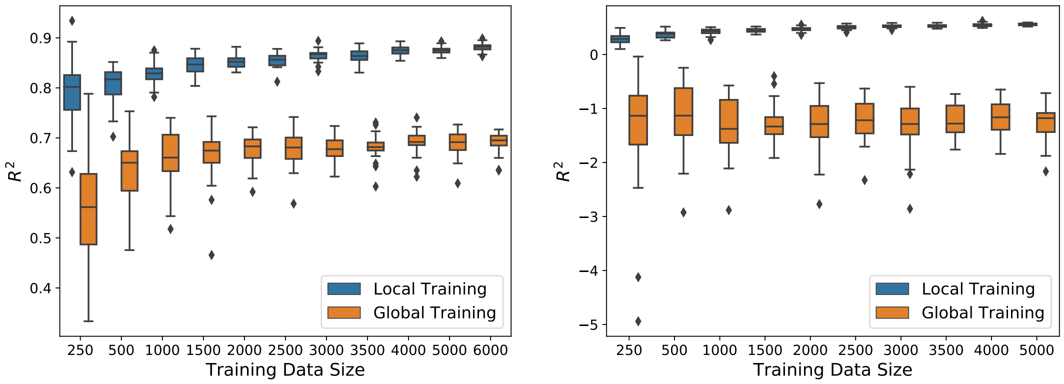
A-D) are presented in Figure 58 and 59. As our main evaluation result, we can state that for all investigated cells, the localized training approach significantly outperformes the common globalized training approach. This behaviour is visible for both transmission directions (upload and download).

This difference in the estimation performance is already noticeable by using a subset of only 1500 measurement points (or even less for some of the cells) and fortifies itself, when using more samples for the training process.

As an additional result, the overall achievable estimation performance clearly varies between the different cells, which further justifies the usage of a location specific set of training data to improve the overall achievable estimation performance.

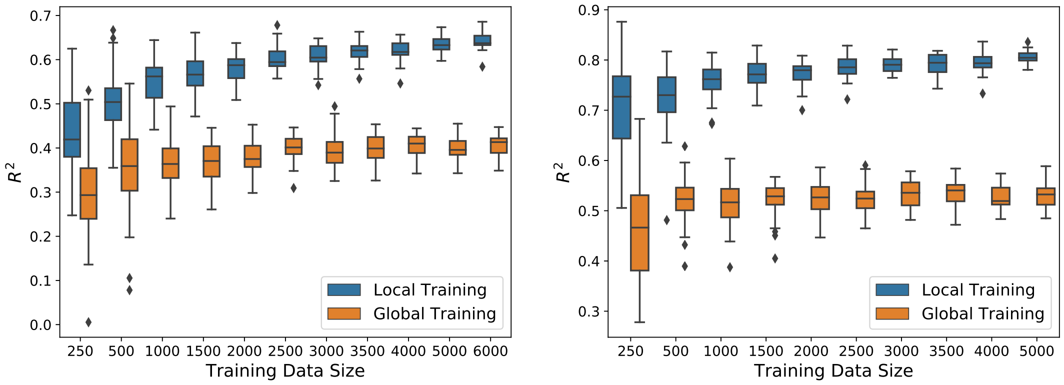
To identify possible reasons for these cell-specific performance results, we investigated the correlation between the used prediction features and the related estimation of the two throughput values on a per cell basis as summarized in Table 16. For a detailed visualization of the achieved correlation parameters for our first and second measuring campaign see Section A.14 in the Appendix.

In our first measuring campaign, where we focused on the investigation of the Android API based throughput estimation, the Reference Signal Received Power (RSRP)



(a) Downlink - Cell C

(b) Uplink - Cell C



(c) Downlink - Cell D

(d) Uplink - Cell D

Figure 59: Continued performance comparison between localized and global training data of the third and fourth most often measured cells of provider A [95].

as one of the measured key performance indicators showed the highest correlation with the estimated throughput values and therefore was considered as our most important estimation feature used by the Random Forest algorithm. For details see Figure 95. With the extended feature set obtained via the MobileInsight application interface during our second measuring campaign further key performance indicators were integrated into our evaluation. To the best of our knowledge we are the first to perform this comparison for such a large set of different network performance parameters. Samba et al. [69] and Yue et al. [155] for example, two of the most recent works, do not consider the availability of Carrier Aggregation as feature for their machine learning process. As shown in Table 16, a different set of performance features could be obtained via MobileInsight for the estimation of the Upload and the Download throughput. This is due to the capabilities of the Nexus 5X smartphone, which we selected as our reference device. The Nexus 5X can only use the technological feature of Carrier Aggregation for the transmission of data in the Downlink. The same holds true for the usage of multiple antennas for the data transmission (MIMO) to save precious battery energy. Consequently the relevance of the different measured parameters for the estimation algorithm not only varies between the different investigated cells, but also for the two

Cells	RSRP	RSRQ	CQI	Carrier Aggregation availability	MIMO availability	Modulation and Coding Scheme
All	-0.012	0.187	0.446	0.6	0.406	0.483
A	0.59	0.342	0.662	-0.00514	0.521	0.599
B	0.804	0.31	0.789	-0.0084	0.526	0.682
C	0.473	0.123	0.42	0.743	0.379	0.46
D	0.279	-0.0177	0.35	0.368	0.248	0.385

(a) Download - Correlation of Throughput with passive measurement values

Cells	RSRP	RSRQ	RSSI	CQI	Modulation and Coding Scheme
All	0.61	0.245	0.576	0.635	0.662
A	0.108	0.0651	0.0877	0.0124	0.294
B	0.564	0.441	0.507	0.584	0.469
C	0.471	0.202	0.446	0.372	0.346
D	0.846	0.223	0.766	0.731	0.76

(b) Upload - Correlation of throughput with passive measurement values

Table 16: Results obtained for the correlation between the achieved throughput and the other passive measured values using Pearson's Correlation Coefficient. The values are showcased for both transmission directions for all the measured cells together (All) and for data, which was only obtained from one of the four most frequently measured cells (A-D) in our dataset [95].

transmission directions (upload, download). When predicting the upload throughput, the Reference Signal Received Power remains a highly important measured feature. This statement holds true when considering the collected samples of all cells, as well as each individual cell's data for the training process. A similar result compared to the evaluation of our first measurement campaign. For details see Figure 99f..

When the Random Forest estimator is trained to predict the download throughput, this correlation changes. Based on the dataset of all four major cells (All), the availability of Carrier Aggregation becomes the most important estimation factor. For details see Figure 96 and 97f.. Supporting our initial proposition to use cell-specific training data, the importance of Carrier Aggregation availability, as a location specific feature, highly varies between individual cells. Its importance is high in the cells C and D along the track, where we measured most of the samples with Carrier Aggregation being present as shown in Figure 54b. In the cells A and B we could only obtain a few samples, where the technology was enabled. Consequently, the Random Forest estimator could not improve its prediction performance through this feature for the specific estimation process conducted in these two cells.

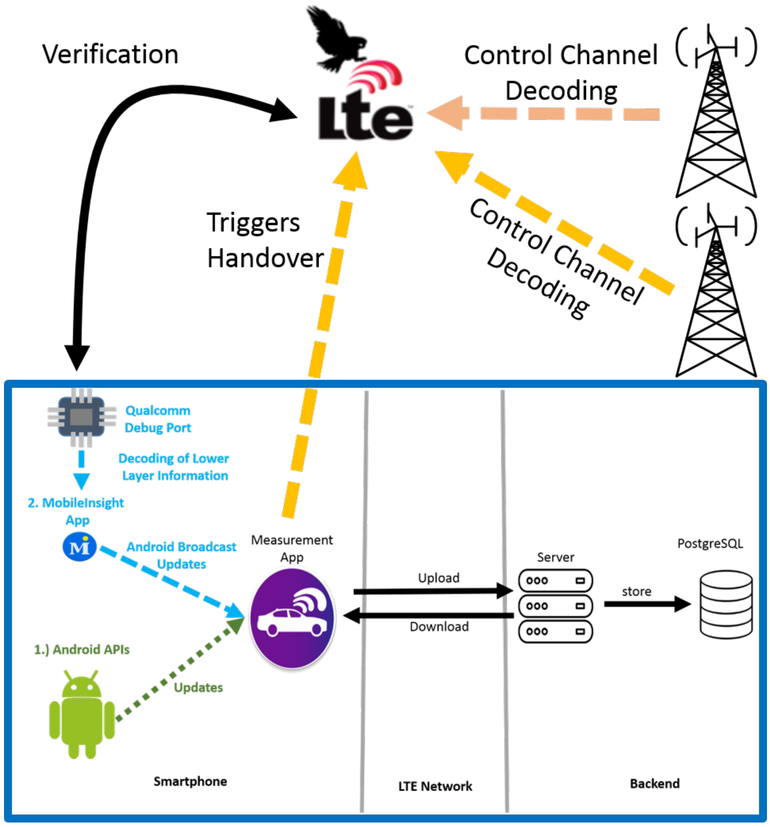


Figure 60: Interconnection between the Connectivity Map Client application and the SDR-based LTE network sniffer Imdea OWL to enable mobile LTE control channel decoding. That way an estimation of the number of active clients in the current cell tower shall be achieved, while the car is traveling.

In conclusion we could showcase through our evaluation that a cell-specific training set significantly improves the performance for the selected Random Forest throughput estimation process.

6.5 MOBILE ESTIMATION OF ACTIVE CLIENTS USING LTE CONTROL CHANNEL DECODING

By combining the information provided by the two previously introduced techniques: Connectivity Map and Online Estimation, self-driving vehicles as our considered mobile network clients, receive detailed information about the network quality in their close proximity and further along their track. Both approaches however cannot provide information about the impact of neighboring active clients (e.g. further vehicles or smartphone users), which might in parallel and independently introduce additional load on the cellular network. To acquire an overview about all currently served clients and their network activities in the range of an LTE cell tower, there exist expensive specialized hardware and software tool sets as stated in Section 3.3.4. These tools realize their functionality through the continuous detailed decoding of the so called Control

Clients have to share cellular network capacity

Challenging application in mobile environment

Channel information, which is continuously transmitted by the LTE cell tower to assign its available network resources each millisecond anew between all its requesting clients. With recent advances in the development of Software Defined Radios, cheap hardware became available, that in combination with custom build decoding software [84, 156–158] is capable to realize similar functionalities. Unfortunately, for our considered mobile application scenario all investigated related and freely available software tool sets [84, 156] of that specific kind have only been tested in a static environment with a fixed distance to the serving cell tower. The main challenge for these so called LTE network sniffers in a mobile application scenario is the continuous change of the currently serving cell tower (executed through the handover process between two adjacent towers in the network) along the path of the vehicle. The currently available tool kits have to be manually reconfigured each time this happens, which renders them unusable for a mobile context. A further difficulty is the resulting varying experienced signal strength. Based on our personal measurements with the LTE network sniffer Imdea OWL [84] we can state that the successful decoding process can only be executed under good signal conditions. In rather poor signal conditions the network sniffer often could not identify a possible neighboring cell, although a smartphone's data connection to the same cell tower was stable. As Imdea OWL is one of the most advanced open source LTE network sniffer, based on our literature research, we decided to investigate and evaluate possible concepts to enable its application in a vehicular environment. To start the decoding procedure of the LTE control channel Imdea OWL initially requires the transmission frequency of the currently serving cell tower as input to setup itself accordingly. Therefore, the network sniffer provides a function to scan the complete spectrum of possible transmission frequencies, to identify the surrounding cell towers. To complete this process however takes up to several minutes. This is feasible for the mentioned static environment as investigated in the Related Work. For our considered mobile application scenario however the cellular environment is changing too frequently, leaving no time for the process to complete with usable results. From our personal measurements conducted in the previously mentioned highway scenarios we can state that a mobile end device such as a smartphone can perform multiple handovers between consecutive cell towers in just some seconds of time, when being transported inside a vehicle (see for example Figure 54a). Our fundamental idea to solve this problem was to enhance the capabilities of Imdea OWL by coupling it with our Android based Connectivity Map Client application as illustrated by Figure 60. The application installed on a smartphone thereby provides Imdea OWL always with the relevant transmission frequency of the smartphone's current serving cell tower. Given this transmission frequency the time intensive initial search process of OWL does not have to be executed. Instead the actual decoding process can directly be started and is completed in only four to five seconds. That way Imdea OWL can be quickly reconfigured to decode another cell tower, when a handover procedure is indicated by the smartphone. The correct decoding performance of Imdea OWL is verified via the MobileInsight application on the smartphone. A correct decoded Imdea OWL result has the same values (C-RNTI and allocated Resource Blocks) as the results provided by MobileInsight directly from the smartphone. This functionality enables various new research applications in our considered mobile scenario. To verify the proposed mechanisms we conducted a first performance evaluation of the described new setup, which is presented in detail in Section A.15.

6.6 FURTHER USAGE OF MEASUREMENT DATA IN SIMULATION

Motivated by our scientific work in the domain of real world cellular network infrastructure we investigated possibilities how our personal contributions could be further applied in the context of vehicular simulations. Not all vehicular communication scenarios can be cost-efficiently tested under real world conditions through a prototypical realization of the proposed concepts. Therefore simulation software provides a reasonable cost-effective alternative. The various currently existing network simulators as introduced in Section 3.3.5 often rely upon complex mathematical models to accurately simulate the different available communication channels. Due to the complexity of these models numerous configuration parameters are required to be setup correctly to achieve reasonable simulations results. This for example includes the assignment of several designated parameters for each access point, as well as the selection of a suitable signal propagation model and its associated parameters to achieve reasonable data rates and latencies of the connection. In consequence the process requires profound background knowledge of the different involved communication technologies (e.g. WLAN and cellular) and especially a steep learning curve to fully understand the used simulation environment.

Our first contribution the RACE Framework as described in the following Section 6.6.1 addresses this exact problem by providing a simple configurable graphical user interface to setup complex vehicular communication scenarios based on real world cellular network environments. The complexity of the underlying simulation models however does not only require human expertise, but also serious compute power and time. This especially holds true if the considered communication scenarios involve hundreds or thousands of communicating vehicles to identify possible scaling effects. Good examples for such situations are our personal simulations conducted in the context of map update distribution as explained in Chapter 4.

To address this general problem of the simulations in terms of required computational power and calculation time we investigated the possibilities of the technical concept of the Connectivity Map as researched upon in several of our own contributions (Sec. 6.1 and 6.3) to be further leveraged in the simulation context, as described in Section 6.6.2.

6.6.1 *Rapid Cellular Network Simulation Framework for Automotive Scenarios (RACE Framework)*

The initial design goal of the Rapid Cellular Network Simulation Framework (RACE Framework) was to create a simulation environment where the applying user could quickly setup and simulate own complex vehicular communication.

Location of one of the cells of a distinct tower (latitude/longitude)
Download and Upload frequencies described by the EARFCN LTE channel numbers
Transmission bandwidth and transmission power
Antenna height, gain, beam width and sending direction angle

Table 17: Parameters to configure each cell of the cell tower infrastructure in ns-3 [96].

Due to our personal obtained background knowledge in the domain of cellular networks, especially the LTE network, as investigated for example in Sec. 6.1 and A.12, we focused our efforts to find a suitable basis for RACE on the investigation of related network simulators that could simulate LTE as physical communication layer.

Easy and detailed configuration, high assistance for the user

To be able to simulate realistic vehicular communication scenarios, we defined three different requirements that our framework had to satisfy: i.) It should be highly configurable in terms of network parameters and network load to closely resemble the cellular network infrastructure as for example experienced in our previous contributions. ii.) It had to provide the possibility to simulate complex vehicular traffic environments, with multiple cars driving around on various tracks. iii.) The framework should assist the human user as much as possible, e.g. through a high degree of automatism where possible, in the scenario creation process. Following the guidance of the framework the user should be able to directly rely upon the created communication scenario for his personal work. No further post processing steps from his side should be required.

By comparing the various available network simulators (Sec. 3.3.5) we identified two promising candidates, the Network Simulator 3 (ns-3) [160] and Omnet++ [159] for the simulation of vehicular communication scenarios.

According to our three stated requirements we selected the ns-3 simulator as most suitable candidate to base our further work upon. At the time of implementation of the RACE framework it provided the more advanced LTE communication stack compared to the one provided by Omnet++.

The currently present LTE stack inside of ns-3 was implemented as part of work of the LENA project[29]. LENA thereby enables a profound configuration of the different cell towers performance criteria, which are summarized in Table 17.

Furthermore LENA was one of the first implementations to simulate the X2 handover procedure of the LTE communication layer. Thereby two different cell towers exchange the serving duty between each other for a unique client. We consider this handover as a highly important feature in our simulations, as it is frequently performed by the roaming vehicles due to their high mobility. This handover feature for example was only integrated later in time in the SimuLTE simulator [166], which provides the fundamental basis for the simulation of LTE networks for the Omnet++ network simulator.

Our second design criteria, the sophisticated vehicular movement simulation, is satisfied by both simulators (ns-3 and Omnet++) by interfacing with the vehicular movement simulator SUMO [106], as already used by us in our previous contributions

---

[29] http://networks.cttc.es/mobile-networks/software-tools/lena/ (Last accessed on August 1, 2019)

RACE enables easy setup of ns-3 LTE stack and SUMO vehicular movement model

(Sec. 4.1, 4.2 and 6.3). Still the creation of cellular communication scenarios in ns-3 requires profound background knowledge and time to correctly setup and configure all required parameters. Thereby the most difficult step in the process is to obtain detailed information about the cellular network infrastructure that shall be simulated. In Germany for example the exact infrastructure of the cellular network is intellectual property of the cellular providers and not publicly available. Some of the required parameters for ns-3 (Tab. 17) such as the cells transmission frequency and bandwidth can be obtained directly from measurement data similar to our collected information (e.g. Tables 9 and 21ff.). Other parameters, such as the location of a cell tower, the installation height of its antennas and the sending angle can be retrieved by cross correlating the measured data with other publicly available data sources, provided by either federal authorities such as the Federal Network Agency (Bundesnetzagentur)[30] or open crowdsourcing projects such as "Cellmapper"[31]. However, the remaining parameters such as the beam width, the transmission power and the gain of the sending antenna require either much more specialized measuring equipment or can only be approximated by commonly used values (e.g. a gain of the sending antenna of 16 dB and a transmission power of about 40 Watts/46 dBm for the cell tower [105]). To solve this major problem the RACE framework instead relies upon a dataset provided by the Canadian Organisation of Innovation, Science and Economic Development (ISED)[32]. In Canada in contrast to Germany the cellular operators are forced by law to provide the ISED with all the information about their cell tower infrastructure. Thus this dataset enables the proper simulation of the whole real world infrastructure of all Canadian network providers, such as Telus, Rogers and Bell[33].

Graphical user interface for scenario creation

To satisfy our third criteria, the high usability, we developed RACE to host a highly intuitive user interface to filter and process the provided data of the ISED's dataset. As first step in the scenario configuration process the user specifies the extend of the simulated area by providing the name of a close by city. Initialized by this information RACE provides a graphical map representation of the cities surrounding area to the user as shown in Figure 61 at the example of the Canadian city of Winnipeg. In this map representation the user then can specify the routes and amounts of the simulated vehicles simply by defining the start and end positions of the various trips by mouse clicks (Fig. 62). Given this information, the RACE framework then selects the celltowers most closely located nearby to the defined routes to be simulated in the ns-3 communication scenario. The resulting scenario provided by the RACE framework then could be directly simulated in ns-3. Fig. 63 for example shows the Radio Environment Map of the cell towers located nearby the example highway scenario of Fig. 62.

The RACE framework is not limited to the ISED's dataset and can be extended in future with additional scenarios, for example via data sets collected from smartphones (e.g. via our Connectivity Map Client application).

---

[30] https://emf.bundesnetzagentur.de (Last accessed on August 1, 2019)

[31] www.cellmapper.net (Last accessed on August 1, 2019)

[32] http:// sms-sgs.ic.gc.ca/eic/site/sms-sgs-prod.nsf/eng/h_00010.html (Last accessed on August 1, 2019)

[33] https://www.ertyu.org/steven_nikkel/cancellsites.html (Last accessed on August 1, 2019)

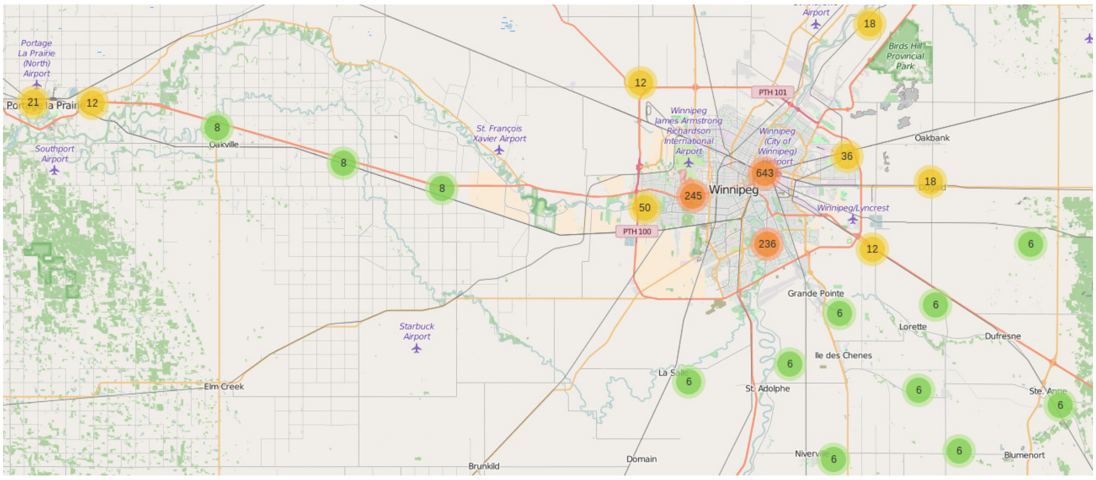


Figure 61: Overview of the LTE infrastructure of provider TELUS Communications Inc. for the city of Winnipeg [96]. Map data ©OpenStreetMap contributors

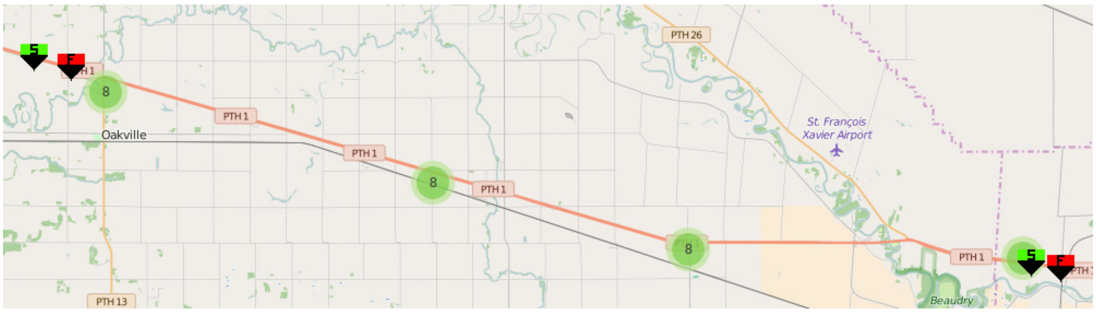


Figure 62: Example highway scenario for the Trans Canada Highway near Winnipeg [96]. Map data ©OpenStreetMap contributors

In conclusion the RACE framework tremendously simplifies the complex task of vehicular communication scenario generation. Still this contribution did not solve the further remaining problem of high requirements of computational power and simulation time for large scale simulations, as it is bound to the LTE simulation model provided by ns-3 through the LENA project. We address this problem with our further contribution in the domain of simulation as described in the following Section 6.6.2.

6.6.2 *An Efficient Heat-Map-Based Wireless Communication Simulation Model for Omnet++*

Very often the testing of the communication capabilities of a specific application does not require the exact simulation of the physical properties of the communication layer through a complex mathematical model (e.g. calculating the transmission signals fading behavior) as it is for example the case for the described ns-3 LTE layer provided by the LENA project. Instead for a proper performance evaluation of the application the simulation of network properties such as throughput and latency on a system

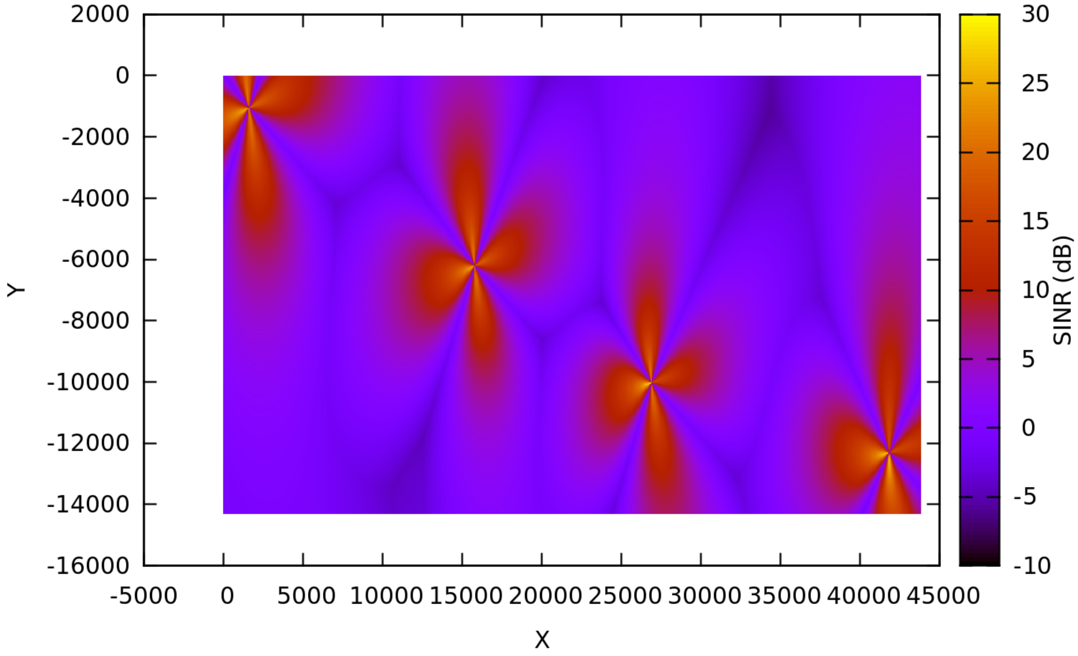


Figure 63: Radio Environment Map plot of the example highway scenario [96].

level perspective is much more important. This especially holds true for vehicular communication scenarios as considered in our personal work (e.g. Sec. 4.1, 4.2 and 6.3), where several hundreds to thousands of vehicles drive in a large area and continuously exchange information between each other. Due to their size such scenarios require immense simulation time or compute power to be evaluated using the existing complex communication models. Consequently we considered possible ways how a further simplification of the present simulation models could be achieved to significantly decrease the required computational power and time necessary for the simulation. As our main contribution we developed the ″Heat-Map-Based Wireless Communication Simulation Model″ and compared it with a state of the art communication model of the network simulation framework Omnet++ INET. We selected Omnet++ as our simulation framework of choice, as we could thereby rely on an already existing communication scenario with known performance values to compare ourselves with [197]. The concept of the Heat Map is thereby inspired from our previous work focusing on the generation of a Connectivity Map. Instead of predicting the future networks performance parameters for a certain vehicle's route by relying on the Connectivity Map we now leverage its contained geo referenced network quality information to describe the networks performance parameters in each area of the simulated scenario. Related to our thoughts Ikuno et al. [198] propose the offline pre-generation of fading parameters for a system-level LTE network simulator. Through this pre-generation the computational time required to simulate the physical LTE layer is also heavily reduced. Consequently Ikuno et al. motivate the feasibility of abstraction for a system-level simulation as considered by us. In contrast to them our personal contribution further abstracts from the used communication technology (e.g. cellular or WLAN). Instead it

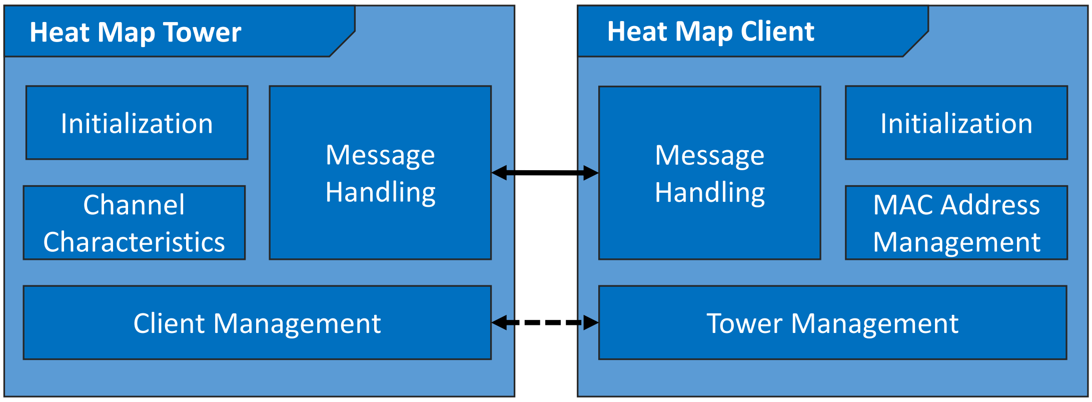


Figure 64: Heat Map modules: tower and client [97]

Heat Map simplifies physical communication layer simulation

simulates the networks performance parameters on a per grid based approach, which we refer to as the Heat Map. By replacing the complex simulation model of Omnet++ INET, the Heat Map model not only achieves a tremendously simplified simulation setup phase, but also heavily improves the required simulation time, as described and evaluated in the following Sections. The design of the transmission model, which enables the wireless communication between the different roaming mobile nodes (e.g. vehicles) and the backend systems (e.g. data processing servers as considered in the Ko-HAF project) is explained in the following Section.

Transmission Model

The Heat Map transmission model is integrated into the ISO OSI layer 2, the data link layer of Omnet++. By handling common Ethernet frames, it enables the usage of all upward internet network protocols without further modification. The routing on the link layer is performed by using MAC addresses. The wireless transmission of all ongoing traffic of one specific network is performed through a central processing base station, the Heat Map tower, that is connected to the backbone network and acts as a transceiver. By initializing several base stations the simulator is able to simulate different wireless networks (e.g. the networks of different cellular providers) through different Heat Maps. To enable the communication, all mobile nodes have to be equipped with a communication counterpart, the Heat Map client. The exact structure of both the Heat Map tower and client is described by Figure 64. The two entities exchange data for the processes of connection and disconnection through their respective management modules (Client Management, Tower Management), as described by the sequence diagram in Figure 65. Therefore the Omnet++ signalling interface, based on common C++ functions, is used to directly exchange the necessary information. In summary the initialisation of clients and their disconnect is handled without transmitting messages over the actual simulated network channel, which significantly improves the simulation's performance. The actual characteristics of the network channel used to transmit the upper layer protocol messages are managed by the Heat Map tower entity

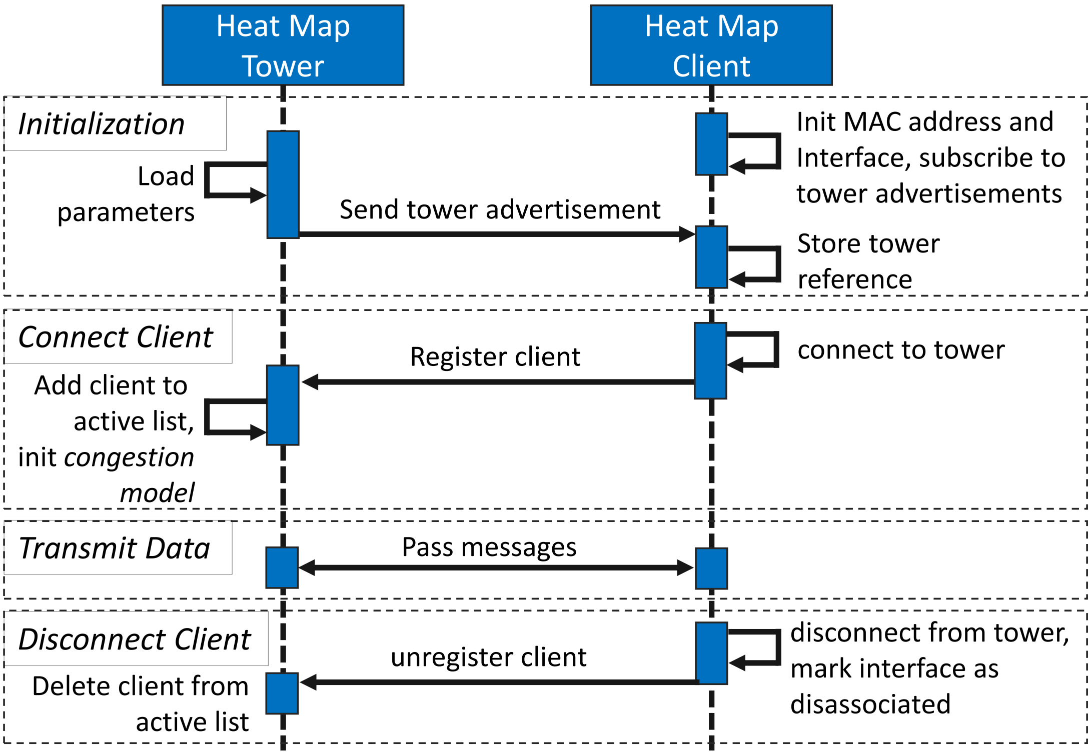


Figure 65: Heat Map Modules - Sequence Diagram [97]

and not as otherwise common by the detailed lower layers of Omnet++. This results in two major advantages of our approach to simulate large scale scenarios, which are only investigated towards their system-level performance. First of all the designers of the simulated scenario do not have to model the complex environmental conditions of the scenario to their full extend to create the desired network characteristics implicitly. Instead the desired parameters such as latency, throughput, drop rate, Maximum Transmission Unit (MTU) and buffer sizes are directly specified through the different layers of the underlying Heat Map. Furthermore this direct specification design allows the network's characteristics to be configured much more precisely for the possible test cases. As second major advantage the simulation time significantly decreases, as the Central Processing Unit (CPU) does not have to compute the complex processes on the lower layers of the Omnet++ INET framework. In our evaluated scenario, as described in the following, we indeed identified that 80% of the total simulation time are spent to perform calculations on these lower layers.

Scenario Creation Process

As introduced at the beginning of this Section our Heat Map model realizes the simulation of network performance characteristics such as the achievable throughput on a per grid level. Therefore the complete simulation scenario is divided into several small grids of equal space. For each grid the described network performance value is the

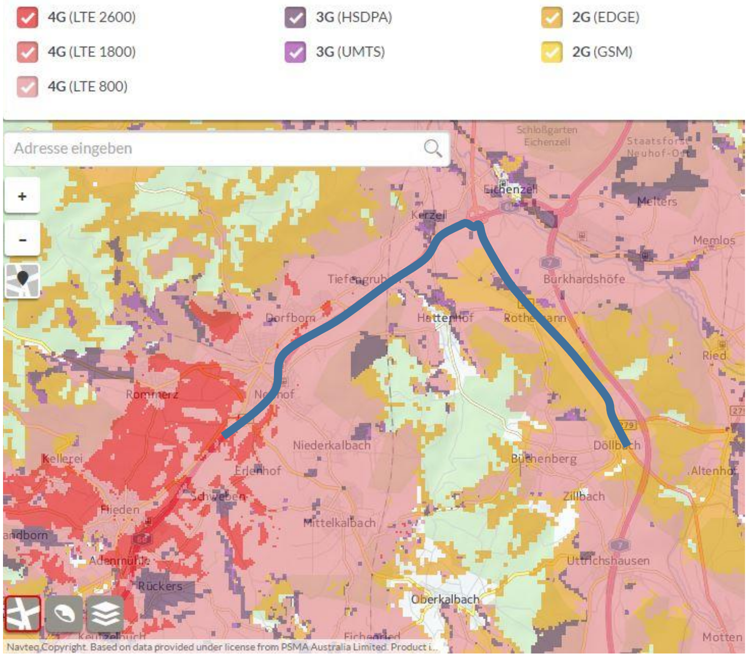


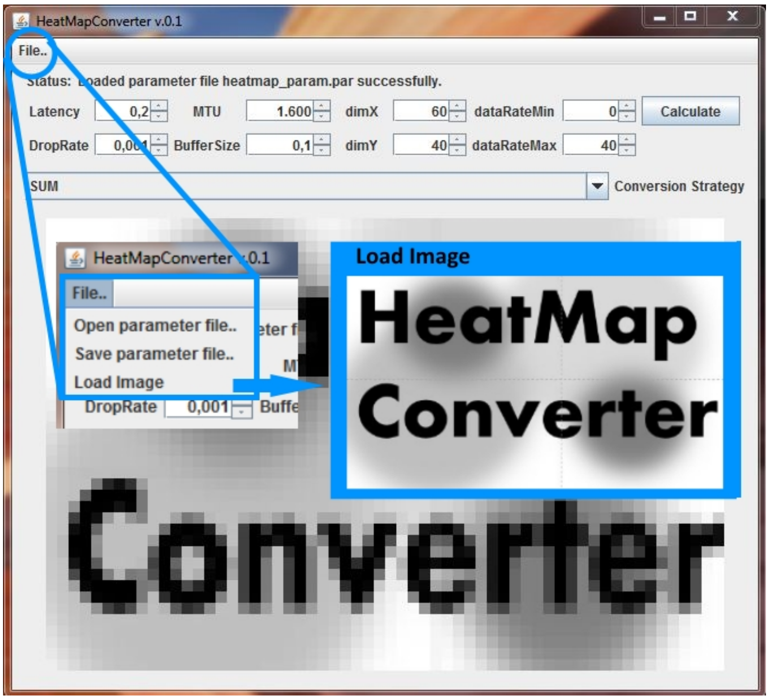
Figure 66: Evaluation Scenario 2: Fulda area, Germany. Extraction from shown Vodafone mobile network availability map [97]. ©Vodafone GmbH & Navteq[34]

same for all clients. In terms of the throughput the Heat Map tower thereby splits the achievable maximum data rates between all active clients in the same grid. The specific size of each grid depends on the required simulation accuracy and the available information about the network. The information regarding the performance of the network for example can be provided by a Connectivity Map as obtained in our previous works (e.g. Sec. 6.1 and 6.2). However the collection of accurate performance measurements as explained in our previous contributions requires significant additional efforts. Consequently data in this highly detailed form might not always be available, but possibly also not required for the considered test scenarios. For certain applications for example it might be sufficient to simulate the large performance changes introduced by the handover between different network technologies (e.g. 2G, 3G and LTE as visualized in Fig. 66 for an example scenario of the German provider Vodafone). Thus to simplify the process of simulation configuration even further we developed a graphical user interface, as shown in Figure 67. With the help of the so called "Heat Map Converter" the user is able to configure the performance of the network by simple specification of different colors in an image. The related performance value of each grid is thereby represented through the color of it's related describing pixels. The Heat Map Converter then transforms these values into a configuration file, in which the performance values of each grid are represented as numeric values. This file is then directly processed by the Heat Map model for the simulation. In the example image of Figure 67 the color black achieves the highest possible throughput, whereas the color white indicates no

GUI to setup individual communication scenarios

---

[34] https://www.vodafone.de/hilfe/netzabdeckung.html (Last accessed on August 1, 2019)



Status: Loaded parameter file heatmap_param.par successfully.

Latency 0,2 MTU 1.600 dimX 60 dataRateMin 0

DropRate 0,001 BufferSize 0,1 dimY 40 dataRateMax 40

SUM Conversion Strategy

Figure 67: Graphical user interface of the Heat Map Converter [97]

network coverage. This visual based design concept allows the user to rely on further publicly available data sources, such as the coverage maps of the cellular providers as shown exemplary in Figure 66 or similar open data sources such as OpenSignal[35] or Cellmapper[36] to create realistic communication scenarios very fast.

Heat Map Evaluation

Reference scenario

For the in-depth performance comparison of our Heat Map communication model with the existing wireless communication layers of the Omnet++ INET framework we selected an already existing WLAN network scenario. The scenario covers an area near Frankfurt am Main, Germany, with motorways, suburban and urban areas involved, as indicated in Figure 68. The simulated network infrastructure consists of nine common WiFi access points simulated in Omnet++ and was previously used to analyse performance characteristics of the mobility management protocol MoVeNet [197]. From this WiFi based simulation we derived our describing Heat Map scenario. From a screenshot of the scenario we derived the location of each access point relatively to each other. Each access point's network characteristics were then described through circular shapes at the identified positions on the Heat Map, which faded out to their specific borders. Each access point thereby describes an independent network and thus nine descriptive Heat Maps were stored separately for the simulation (one for each of the access points). As following step we generated the distinct Heat Map configuration

---

[35] http://www.opensignal.com (Last accessed on August 1, 2019)

[36] http://www.cellmapper.net (Last accessed on August 1, 2019)

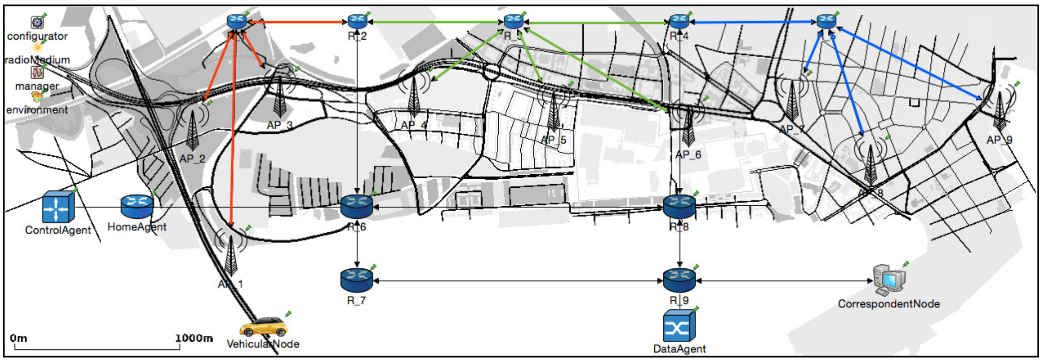


Figure 68: Evaluation Scenario 1: Frankfurt am Main, Germany. Redefinition of an existing scenario with Heat Map tools [97]

files via the Heat Map Converter using the nine given image sources. That way the whole setup process of the Heat Map based simulation could be completed in under five minutes. For the evaluation of the Heat Map model we compare its runtime performance with the corresponding runtime of the default Omnet++ INET WiFi models for our chosen scenario. We execute the simulation with one single mobile node representing a transmitting vehicle and vary its sending data rate (the amount of transmitted data) throughout the evaluation. Similar to our previous evaluation (e.g. Section 4.1.5) each simulation run is repeated 30 times to ensure the significance of the achieved evaluation results. Throughout each simulation run UDP packets are send in both transmission directions (upload and download). Thereby we varied the achieved data rates between 1 Mbit/s and 64 Mbit/s to investigate the different model performances under different amounts of network load. 64 Mbit/s was selected as upper boundary of the evaluated data rate to experience the effects of the saturation of the wireless channel. Figure 69 represents the achieved performance of the Heat Map model as a black line.

The Omnet++ WiFi reference model is indicated by a blue dashed line. Both axis of the graph have a logarithmic scale. Thus we observe that both models achieve a linear correlation between the number of packets and the required computational time of the simulation. The required simulation time doubles for the Omnet++ WiFi model, when quadrupling the number of transmitted packets. The Heat Map model achieves a significant better performance, with initially comparably low simulation times. These times double when the network time is doubled. In consequence both lines converge at one point, where the simulation of the upper layers outweighs the simulation of the physical layer. We analyze the absolute gain in simulation time savings to investigate this effect in depth. This absolute gain is indicated by the magenta dashed line in Figure 69. It shows that the simulation time gain saturates correlating to the wireless channels saturation at a data rate of about 16 Mbit/s. The convergence behavior between the two investigated models is shown by the percentage-based performance gain, indicated by the cyan stems in the graph of Figure 69. The scale of the percentage-based performance gain is linear and associated with the right side of the figure. For the simulated data

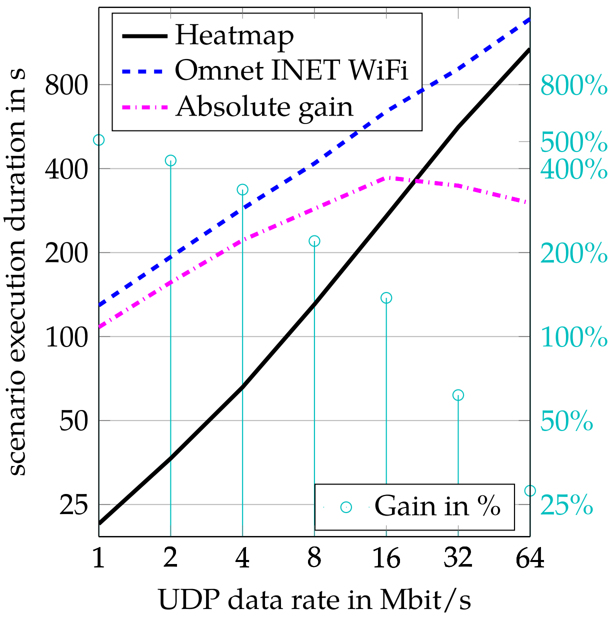


Figure 69: Simulation duration comparison over data rate [97]

rate of 1 Mbit/s we observe a gain of approximately 500% in simulation time between the common WiFi model and the Heat Map model. In consequence more than 80% of the simulation time of this scenario was required by the CPU to calculate lower layer processes. Doubling the transmitted data rate to 2 Mbit/s reduces this gain less than linear. The performance gain then converges to zero with the further increase of the data rate. This is due to the saturation of the wireless channel at around 16 Mbit/s. After this point any additional packets are dropped at the wireless interface. However the rest of the networks layers still fully processes these packets, which leads to an increase. Consequently the impact of the Heat Map model on the overall simulation time decreases, which then leads to the identified convergence of the two compared models.

In summary we could show a significant advantage of the Heat Map model over the common Omnet++ WiFi model in terms of simulation time (up to 500%), especially in scenarios where the wireless transmission is the most computation intensive part of the simulation.

7

SUMMARY, CONCLUSIONS, AND OUTLOOK

To conclude our work, we summarize the content of the previous Chapters and state our main contributions in the following. We then draw conclusions based on our obtained results. Finally, we discuss open issues and potential future work.

7.1 SUMMARY OF THE THESIS

In Chapter 1, we motivated the benefits of a High Definition Street Map (HD Map) for self-driving vehicles. The HD Map provides highly precise geographic referenced location information to improve the overall driving safety and comfort of the cars. Following this motivation we described in detail the research challenges related to the necessary, continuous maintenance of the HD Map using wireless communication links as the focus of this thesis. Namely they are *i) the Reliable Distribution of Map Data Updates* for all self-driving vehicles ii) a *Fast Road Infrastructure Change Detection* to incorporate such areas as updates into the map and maintain its usability over time as well as iii) the *Reliable Provision Via Wireless Communication* of those two data streams (map updates and vehicular sensor data) towards and from the self-driving vehicles. Based on our description of related background information in Chapter 2 and our analysis of the state of the art in Chapter 3 we presented the following contributions in our thesis.

7.1.1 *Contributions and Conclusions*

The presented work identified research challenges in the three major building blocks (Distribution, Generation and Provision) of the maintenance cycle of the High Definition Map (HD Map). Our proposed contributions to resolve them and the drawn conclusions are grouped accordingly in the following.

Reliable and Efficient HD Map Update Distribution for Self-Driving Vehicles

Following our study of Related Work regarding existing concepts of partial and incremental map updates we identified further optimization potential in the distribution process of the map updates considering the requirements and prerequisites of self-driving vehicles. In consequence we proposed our first contribution the *Dynamic Map Update protocol* [170] as described in Chapter 4. The protocol enables the context specific selection and wireless transmission of mandatory map data from a central backend entity (e.g. a server) to all requesting self-driving vehicles via the cellular network.

Context specific map update selection

Ad hoc map data sharing

Additionally to the centralized update mechanism we conducted experiments on the impact of WLAN-based vehicular ad hoc communication to distribute already received map updates between the vehicles [88] themselves. Therefore we extended the initial specification of the Dynamic Map Update protocol to further reduce the otherwise necessary and costly cellular network traffic, as well as the processing load on the central map maintaining backend entity. We evaluated the performance of the protocol using the traffic simulator SUMO on actual map data of the German city of Berlin and the area of Luxembourg. Based on our evaluation results we showed that the distribution process of map updates can be significantly optimized via an intelligent context specific preselection of mandatory and optional updates. Thereby we achieved the same effectiveness in road guidance with only a fraction of transmitted map updates compared to related existing concepts. We furthermore showcased the positive impact of ad hoc communication on the distribution process by offloading necessary data transmissions from the cellular network, which further reduced the processing load on the central server, as well as the overall transmissions costs.

Fast and Reliable Detection of Changes in the Road Infrastructure Based on Low-cost Widely Distributed Sensor Data

Fast detection of changes through ubiquitous sensors

In the research context of the application of vehicular sensor data for the maintenance of the HD Map we identified a research gap in the detection of changes in the road network. We addressed this gap by our *lane course change detection algorithm* [89] presented in Chapter 5. To achieve a fast and reliable detection of changes, the algorithm leverages low-cost sensor information (GNSS location information, as well as accelerometer and gyroscope readings) from ubiquitously deployed aftermarket end devices (e.g. smartphones). At the example of a real highway construction site scenario we verified that our approach achieves a fast and reliable detection of changed lane courses using only the described low-cost sensors, which are integrated into smartphones. The algorithm clearly designated the construction area that changed over time with a sub-lane-level accuracy. This accurate detection showed the great potential of ubiquitously available after market devices to enhance the generation speed of HD Map updates. The intelligent combination of the different sensor sources was missing in Related Work, which either relied upon expensive sensor equipment or only investigated the benefits of one sensor source at a time.

Robust, Optimized Provisioning of Data w.r.t Changing Environmental Conditions

As major research focus of our thesis we investigated techniques to optimize the wireless provision of the previously mentioned data (HD Map updates and vehicular sensor data) for the maintenance of the HD Map using the currently established cellular network infrastructure as described in Chapter 6. We focused our research efforts on the intelligent and data efficient collection of performance measurements of the cellular network and the prediction of the related key performance indicators. That way a broad overview about the network quality parameters in close and far proximity to the vehicle's current track is obtained. This valuable information then is assumed to be forwarded to existing related scheduling algorithms, which incorporate it to optimize their data transmission planning. This planning process itself is out of the focus of this thesis.

As initial contribution we analyzed the *communication requirements of self-driving cars* based on actual prototype vehicles [92]. To put the vehicle's requirements in correlation to the surrounding vehicular traffic and the cellular network's capacity limits we conducted a *requirements analysis* as well as an *extensive measuring campaign* of the deployed LTE network infrastructure on various highways and country roads around Frankfurt am Main, Germany.

Especially the initially assumed challenging high variance in the performance of the cellular network due to various geographic influence factors could be verified. Furthermore we identified a high demand of the self-driving cars towards the upload bandwidth required for the transmission of vehicular sensor data. This demand stands in contrast to the provided capacities of the present cellular network infrastructure, which are much higher in the download direction compared to the upload.

Network performance vs vehicular requirements

To maintain this gained temporarily knowledge about the quality of the cellular network over a longer period of time and especially larger areas we developed the Intelligent Cellular Communication Quality Sensing framework (ICCOMQS). ICCOMQS enables the data-efficient participation of all self-driving vehicles in the measuring and sharing of network quality and performance parameters via a distinct layer in the HD Map. The framework achieves this performance by intelligently scheduling the anyway necessary data transmissions (e.g. map updates and vehicular sensor data) to execute measurements in areas where no information about the performance of the cellular network currently is available. In consequence no further costly dummy data for the probing of the network is required anymore. That way the framework achieves the same measuring accuracy as related active probing approaches at the level of data-efficiency of related passive probing approaches. Through an intelligent selection of future measuring points ICCOMQS furthermore achieves a much better coverage of area in comparison to the immediate sending of data.

Intelligent cooperative probing of cellular network

As following contribution the capabilities of the framework, which were initially evaluated through simulation, were implemented in an *actual prototype*. The prototype was realized via a smartphone based client and a Linux based server. Using this setup we conducted additional series of measurements and collected a further extensive set of geo-referenced cellular network performance indicators. The measuring results obtained from the common smartphones were similar to the ones collected from the fleet of prototype vehicles and furthermore encouraging comparable to results of a reference measurement conducted by a German provider. These correlations showcase that a broad knowledge about the performance of the surrounding cellular network can be obtained and shared between vehicles and smartphones a like.

Measuring campaigns

Using the dataset obtained via the ICCOMQS prototype we enhanced the vehicle's short term knowledge (up to several seconds of time) about the experienced surrounding network to further improve the data transmission. Therefore we leveraged the collected network quality measurements to exemplary enhance an existing machine-learning based throughput prediction algorithm by training it on sets of exactly geo-located data [94, 95]. The obtained evaluation results outperformed comparable approaches in the Related Work, which commonly rely on one single training data set and therefore do not consider the location as an important training criteria.

Online prediction via geo-located training data

Furthermore we enabled the application of the ImdeaOWL LTE control channel decoder in a moving vehicle by coupling it with our smartphone based ICCOMQS framework. That way even further detailed information about the current capacity and the amount of active clients in the LTE network can be obtained via test drives. Previously this information could only be obtained in a stationary position.

Mobile LTE control channel decoding

In summary we improved the combined provision of long term and short term cellular network quality information to be used by scheduling algorithms inside the self-driving vehicle for an efficient and reliable data transmission.

As our final contribution we investigated further concepts to leverage the obtained data set of cellular network performance parameters in simulations. As result we contributed the Rapid Cellular Simulation framework (RACE), that in comparison to related simulation frameworks allows a much easier and quicker setup of realistic vehicular communication scenarios. Additionally we implemented and evaluated a communication model, which directly incorporates actual measured cellular performance parameter in its own configuration. It outperformed an existing mathematical communication model in terms of run time and thus is suited for large scale scenarios, which could otherwise not be simulated in a reasonable amount of time.

Vehicular communication frameworks and models

7.2 OUTLOOK

The contributions and achieved results presented in this thesis motivate further research upon the ongoing optimization and improvement of the High Definition Map, as commonly shared knowledge base of self-driving vehicles. Even with an expected steady increase of automated driving capabilities in vehicles over the upcoming years [199, 200] there will be always ″extreme″ situations, where the car's artificial intelligence fails to select a suitable driving maneuver on its own. The High Definition Map as presented in this thesis helps to solve such situations, in which for example a human driver also would be uncertain (e.g. "Is the water shallow enough to safely drive through?"), to ensure a high level of safety and comfort. For a proper maintenance of the map we especially consider the verification and aggregation of the various sensor sources provided by different vehicles as a challenging task for future research. To aggregate different sensor readings into a correct and verified model of the ongoing traffic environment requires further research. The generation and processing of the required data streams for self-driving vehicles also motivates further research in the

A priori knowledge via HD Map

Sensor data aggregation

domain of wireless communication to fast and reliably exchange data. In our opinion the most relevant performance parameter for the self-driving vehicle are the latency and throughput that can be achieved via a communication channel. In consequence related research areas such as ad hoc communication [26], tele operation [201], edge computing [202] or beamforming [203–205], provide large research potential in the application context of self-driving vehicles. In summary we consider any additional sensor and technology, that provide further reliable information to a self-driving vehicle to make safe and reasonable driving decisions a worthwhile consideration for future research.

Reliable data exchange

ACKNOWLEDGMENTS

This work has been founded by and executed on behalf of the Opel Automobile GmbH and the German Federal Ministry of Economics as part of the German research project Ko-HAF (Cooperative Highly Automated Driving).

[1] Daniel Sperling and Deborah Gordon. *Two Billion Cars: Driving Toward Sustainability*. Google-Books-ID: 8IM8DwAAQBAJ. Oxford University Press, July 22, 2010. 325 pp. ISBN: 978-0-19-973723-9.

[2] VDA - Verband der Automobilindustrie. "Von Fahrerassistenzsystemen zum automatisierten Fahren." In: *VDA Magazin - Automatisierung* (2015). URL: `https://www.vda.de/dam/vda/publications/2015/automatisierung.pdf` (Last accessed on August 1, 2019).

[3] World Health Organization. "Mobile Phone Use: A Growing Problem of Driver Distraction." In: (2011). URL: `https://www.who.int/violence_injury_prevention/publications/road_traffic/distracted_driving_en.pdf` (Last accessed on August 1, 2019).

[4] Statistisches Bundesamt Deutschland. *Unfallentwicklung auf deutschen Strassen 2017*. 2018. (Last accessed on August 1, 2019).

[5] Fabian Kröger. "Automated Driving in Its Social, Historical and Cultural Contexts." In: *Autonomous Driving: Technical, Legal and Social Aspects*. Ed. by Markus Maurer, J. Christian Gerdes, Barbara Lenz, and Hermann Winner. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 41–68. ISBN: 978-3-662-48847-8. DOI: `10.1007/978-3-662-48847-8_3`. URL: `https://doi.org/10.1007/978-3-662-48847-8_3` (Last accessed on August 1, 2019).

[6] Richard Viereckl, Dietmar Ahlemann, Alex Koster, and Sebastian Jursch. "Racing Ahead with Autonomous Cars and Digital Innovation." In: *Auto Tech Review* 4.12 (Dec. 2015), pp. 18–23. ISSN: 2250-3390. DOI: `10.1365/s40112-015-1049-8`. URL: `http://link.springer.com/10.1365/s40112-015-1049-8` (Last accessed on August 1, 2019).

[7] Alexandria Sage. *Waymo unveils self-driving taxi service in Arizona for paying customers*. URL: `https://www.reuters.com/article/us-waymo-selfdriving-focus/waymo-unveils-self-driving-taxi-service-in-arizona-for-paying-customers-idUSKBN1041M2` (Last accessed on August 1, 2019).

[8] Julius Ziegler, Philipp Bender, Markus Schreiber, and Henning et al. Lategahn. "Making Bertha Drive - An Autonomous Journey on a Historic Route." In: *IEEE Intelligent Transportation Systems Magazine*. Vol. 6. 2. 2014, pp. 8–20. DOI: `10.1109/MITS.2014.2306552`. URL: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6803933` (Last accessed on August 1, 2019).

[9] Jonathan M. Gitlin. *The most detailed maps of the world will be for cars, not humans*. Ars Technica. 2017. URL: `https://arstechnica.com/cars/2017/03/the-most-detailed-maps-of-the-world-will-be-for-cars-not-humans/` (Last accessed on August 1, 2019).

[10] Heiko G. Seif and Xiaolong Hu. "Autonomous Driving in the iCity—HD Maps as a Key Challenge of the Automotive Industry." In: *Engineering*. Vol. 2. 2. 2016, pp. 159–162.

[11] Alexis C. Madrigal. *The Trick That Makes Google's Self-Driving Cars Work - The Atlantic - GOOGLE*. 2014. URL: `http://www.theatlantic.com/technology/archive/2014/%2005/all-the-world-a-track-the-trick-that-makes-googles-self-driving-cars-work/%20370871/` (Last accessed on August 1, 2019).

[12] Christopher Lawton. *Why an HD map is an essential ingredient for self-driving cars - HERE 360, http://360.here.com/2015/05/15/hd-map-will-essential-ingredient-self-driving-cars/*. 2015. (Last accessed on August 1, 2019).

[13] Alexis C. Madrigal. *The Trick That Makes Google's Self-Driving Cars Work - The Atlantic*. 2014. URL: `https://www.theatlantic.com/technology/archive/2014/05/all-the-world-a-track-the-trick-that-makes-googles-self-driving-cars-work/%20370871/` (Last accessed on August 1, 2019).

[14] Jocelyn Plack. *The unmatched quality of HERE Maps content - HERE 360, http://360.here.com/2013/03/27/the-unmatched-quality-of-here-maps-content/*. 2013. (Last accessed on August 1, 2019).

[15] Florian Jomrich. "Intelligent Provisioning of High Definition Street Maps for Highly Automated Driving Vehicles." English. In: *Proceedings of: International Conference on Networked Systems (NetSys 2017)*. Mar. 2017, p. 2. URL: `netsys17.uni-goettingen.de/wp-content/uploads/2017/03/phdforum-paper15.pdf`.

[16] Jamie Stevenson. *Making the invisible visible with the HD Live Map*. 2016. URL: `http://360.%20here.com/2016/09/23/making-the-invisible-visible-with-the-hd-live-map-video-demo/` (Last accessed on August 1, 2019).

[17] TomTom. *TomTom Enables Autonomous Driving*. 2017. URL: `http://wonderlanddev.com/%20tomtomv2/wordpress/wp-content/uploads/2017/01/Autonomous-Driving-%20Product-Info-Sheet.pdf,%20[online%20accessesd%202019-05-15]` (Last accessed on August 1, 2019).

[18] Dr. Nicole Beringer. "Sensor-based learning algorithms pave the way towards autonomous driving." In: *87th Automotive Meets Electronics*. VDE Verlag, 2017, pp. 26–31.

[19] Thomas Barthel. *Map data as co-pilot: Extended scope for autonomous driving*. 2017. URL: `http://www.embedded-computing.com/embedded-computing-design/map-data-as-co-pilot-extended-scope-for-autonomous-driving` (Last accessed on August 1, 2019).

[20] Parkinson, Bradford W., Enge, Per, Axelrad, Penina, and Spilker Jr., James J. *Global Positioning System: Theory and Applications, Volume II - Progress in Astronautics and Aeronautics*. merican Institute of Aeronautics and Astronautics, 1996. ISBN: 978-1-56347-107-0. URL: `https://arc.aiaa.org/doi/abs/10.2514/4.866395`.

[21] Kenji Ishikawa, Michima Ogawa, Shigetoshi Azuma, and Tooru Ito. “Map navigation software of the electro-multivision of the’91 Toyoto Soarer.” In: *Vehicle Navigation and Information Systems Conference, 1991*. Vol. 2. IEEE, 1991, pp. 463–473. URL: `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1623657` (Last accessed on August 1, 2019).

[22] Antony Cooper and Ammatzia Peled. “Incremental updating and versioning.” In: *Proceedings of 20 th International Cartographic Conference*. 2001, pp. 2804–2809. URL: `http://icaci.org/files/documents/ICC_proceedings/ICC2001/icc2001/file/f19007.pdf` (Last accessed on August 1, 2019).

[23] Yan Liu, Zhaosheng Yang, and Xiufeng Han. “Research for Incremental Update Data Model applied in Navigation Electronic Map.” In: *Advanced Computer Control (ICACC), 2010 2nd International Conference on*. Vol. 1. IEEE, 2010, pp. 261–265. URL: `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5487031` (Last accessed on August 1, 2019).

[24] Kyoungwook Min. “A System Framework for Map Air Update Navigation Service.” In: *ETRI Journal*. Vol. 33. 4. 2011, pp. 476–486. DOI: `10.4218/etrij.11.1610.0012`. URL: `http://etrij.etri.re.kr/Cyber/BrowseAbstract.jsp?vol=33&num=4&pg=476` (Last accessed on August 1, 2019).

[25] *Navigation System Map Update - Instruction Guide*. 2018. URL: `https://gmnavdisc.navigation.com/web/WFS/Shop-GMNA-Site/en_US/-/USD/%20ViewProductAttachment-OpenFile?LocaleId=en_US&DirectoryPath=GMNA&FileName=GM+NGI+2.5+WEB_Guide+2018.pdf&UnitName=Shop` (Last accessed on August 1, 2019).

[26] N. Lu, N. Cheng, N. Zhang, X. Shen, and J. W. Mark. “Connected Vehicles: Solutions and Challenges.” In: *IEEE Internet of Things Journal* 1.4 (Aug. 2014), pp. 289–299. ISSN: 2327-4662. DOI: `10.1109/JIOT.2014.2327587`.

[27] He Ling. *NavInfo: Incremental Update is the Revolution for Map Production System*. 2013. URL: `http://www.navinfo.com/en/news/detail.aspx?id=944%20%5C&sort=3` (Last accessed on August 1, 2019).

[28] Automotive Ltd. Hitachi. *Map update service/solution*. 2016. URL: `https://web.archive.org/%20web/20150822121843/http://www.hitachi-automotive.co.jp/en/products/cis/02.html` (Last accessed on August 1, 2019).

[29] Akinori Asahara, Masaaki Tanizaki, Michio Morioka, and Shigeru Shimada. “Locally Differential Map Update Method with Maintained Road Connections for Telematics Services.” In: *Mobile Data Management Workshops, 2008. MDMW 2008. Ninth International Conference on*. IEEE, 2008, pp. 11–18. URL: `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4839079` (Last accessed on August 1, 2019).

[30] ETSI. *ETSI TR 102 863, Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Local Dynamic Map (LDM); Rationale for and guidance on standardization*. 2011. (Last accessed on August 1, 2019).

[31] Hideki Shimada, Akihiro Yamaguchi, Hiroaki Takada, and Kenya Sato. "Implementation and Evaluation of Local Dynamic Map in Safety Driving Systems." In: *Journal of Transportation Technologies* 05.2 (2015), pp. 102–112. ISSN: 2160-0473, 2160-0481. DOI: `10.4236/jtts.2015.52010`. URL: `http://www.scirp.org/journal/PaperDownload.aspx?DOI=10.4236/jtts.2015.52010` (Last accessed on August 1, 2019).

[32] Hamid Menouar, Fethi Filali, and Adnan Abu-Dayya. "Efficient and unique identifier for V2X events aggregation in the local dynamic map." In: *ITS Telecommunications (ITST), 2011 11th International Conference on*. IEEE, 2011, pp. 369–374. URL: `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6060084` (Last accessed on August 1, 2019).

[33] Eric Koenders, Dick Oort, and Klaas Rozema. "An open Local Dynamic Map." In: (Aug. 2, 2015). URL: `http://www.compass4d.eu/download/eks_itseu2014_an_open_local_dynamic_map.pdf` (Last accessed on August 1, 2019).

[34] Christian Ress, Dirk Balzer, Alexander Bracht, Sinisa Durekovic, and Jan Löwenau. "Adasis protocol for advanced in-vehicle applications." In: *15th World Congress on Intelligent Transport Systems*. 2008, p. 7. URL: `http://www.cvt-project.ir/en/Admin/Files/eventAttachments/ADASISv2-ITS-NY-Paper-Finalv4_149.pdf` (Last accessed on August 1, 2019).

[35] Daniel Burgstahler, Athiona Xhoga, Christoph Peusens, Martin Möbus, Doreen Böhnstedt, Ralf Steinmetz, and Firstname Lastname. "RemoteHorizon.KOM: Dynamic Cloud-based eHorizon." In: (2016), p. 6.

[36] Linda See, Peter Mooney, Giles Foody, Lucy Bastin, Alexis Comber, Jacinto Estima, Steffen Fritz, Norman Kerle, Bin Jiang, Mari Laakso, Hai-Ying Liu, Grega Milčinski, Matej Nikšič, Marco Painho, Andrea Pődör, Ana-Maria Olteanu-Raimond, and Martin Rutzinger. "Crowdsourcing, Citizen Science or Volunteered Geographic Information? The Current State of Crowdsourced Geographic Information." In: *ISPRS International Journal of Geo-Information* 5.5 (Apr. 27, 2016), p. 55. ISSN: 2220-9964. DOI: `10.3390/ijgi5050055`. URL: `http://www.mdpi.com/2220-9964/5/5/55` (Last accessed on August 1, 2019).

[37] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava. "Participatory sensing." In: *In: Workshop on World-Sensor-Web (WSW'06): Mobile Device Centric Sensor Networks and Applications*. 2006, pp. 117–134.

[38] Jeff Howe. "The Rise of Crowdsourcing." In: *Wired Magazine* 14 (2006), p. 5.

[39] Georg Gartner, David A. Bennett, and Takashi Morita. "Towards Ubiquitous Cartography." In: *Cartography and Geographic Information Science* 34.4 (2007), pp. 247–257. ISSN: 1523-0406, 1545-0465. DOI: `10.1559/152304007782382963`. URL: `http://www.tandfonline.com/doi/abs/10.1559/152304007782382963` (Last accessed on August 1, 2019).

[40] Werner Huber, Michael Lädke, and Rainer Ogger. "Extended Floating Car Data for the Acquisition of Traffic Information." In: *Proceedings of the ITS-World Congress* (1999), p. 9.

[41] Stefan Schroedl and et al. "Mining GPS traces for map refinement." In: *Data mining and knowledge Discovery* 9.1 (2004), pp. 59–87. URL: `http://link.springer.com/article/10.1023/B:DAMI.0000026904.74892.89` (Last accessed on August 1, 2019).

[42] Stefan Edelkamp and Stefan Schroedl. "Route planning and map inference with global positioning traces." In: *Computer Science in Perspective*. Springer, 2003, pp. 128–151. (Last accessed on August 1, 2019).

[43] Xuemei Liu and et al. "Mining large-scale, sparse GPS traces for map inference: comparison of approaches." In: *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012, pp. 669–677. URL: `http://dl.acm.org/citation.cfm?id=2339637` (Last accessed on August 1, 2019).

[44] Chunzhao Guo, Kiyosumi Kidono, Junichi Meguro, Yoshiko Kojima, Masaru Ogawa, and Takashi Naito. "A Low-Cost Solution for Automatic Lane-Level Map Generation Using Conventional In-Car Sensors." In: *IEEE Transactions on Intelligent Transportation Systems* 17.8 (2016), pp. 2355–2366. ISSN: 1524-9050, 1558-0016. DOI: `10.1109/TITS.2016.2521819`. URL: `http://ieeexplore.ieee.org/document/7422083/` (Last accessed on August 1, 2019).

[45] Weidong Fang, Rong Hu, Xiang Xu, Ye Xia, and Mao-Hsiung Hung. "A novel road network change detection algorithm based on floating car tracking data." In: *Telecommunication Systems* (2016). ISSN: 1018-4864, 1572-9451. (Last accessed on August 1, 2019).

[46] Robert Neuhold, Michael Haberl, Martin Fellendorf, Gernot Pucher, Mario Dolancic, Martin Rudigier, and Jörg Pfister. "Generating a Lane-Specific Transportation Network Based on Floating-Car Data." In: *Advances in Human Aspects of Transportation*. Ed. by Neville A. Stanton, Steven Landry, Giuseppe Di Bucchianico, and Andrea Vallicelli. Vol. 484. Cham: Springer International Publishing, 2017, pp. 1025–1037. ISBN: 978-3-319-41681-6. (Last accessed on August 1, 2019).

[47] Heba Aly and et al. "Lanequest: An accurate and energy-efficient lane detection system." In: *Pervasive Computing and Communications (PerCom), 2015 IEEE International Conference on*. IEEE, 2015, pp. 163–171. (Last accessed on August 1, 2019).

[48] Dongyao Chen and et al. "Invisible Sensing of Vehicle Steering with Smartphones." In: ACM Press, 2015, pp. 1–13. ISBN: 978-1-4503-3494-5. DOI: `10.1145/2742647.2742659`. URL: `http://dl.acm.org/citation.cfm?doid=2742647.2742659` (Last accessed on August 1, 2019).

[49] Zhichen Wu and et al. “L3: Sensing driving conditions for vehicle lane-level localization on highways.” In: *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*. IEEE, 2016, pp. 1–9. (Last accessed on August 1, 2019).

[50] Yan-ling Deng and et al. “An Algorithm for Identifying the Change of Road Restriction Based on K-means Clustering.” In: *DEStech Transactions on Computer Science and Engineering* (mcsse 2016). URL: `http://www.dpi-proceedings.com/index.php/dtcse/article/view/10949` (Last accessed on August 1, 2019).

[51] Umama Ahmed and et al. “Minimizing GPS Dependency for a Vehicle’s Trajectory Identification by Using Data from Smartphone Inertial Sensors and Onboard Diagnostics Device.” In: *Transportation Research Record: Journal of the Transportation Research Board* 2644 (2017), pp. 55–63.

[52] Icaro Da Silva, Salah Eddine El Ayoubi, Orange Mauro Boldi, Ömer Bulakci, Panagiotis Spapis, Malte Schellmann, Jose F. Monserrat, Thomas Rosowski, Gerd Zimmermann, Deutsche Telekom, et al. “5G RAN Architecture and Functional Design.” In: (2016).

[53] S. Onoe. “1.3 Evolution of 5G mobile technology toward 1 2020 and beyond.” In: *2016 IEEE International Solid-State Circuits Conference (ISSCC)*. IEEE, 2016, pp. 23–28. URL: `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7417891` (Last accessed on August 1, 2019).

[54] S. Dimatteo, P. Hui, B. Han, and V. O. K. Li. “Cellular Traffic Offloading through WiFi Networks.” In: *2011 IEEE Eighth International Conference on Mobile Ad-Hoc and Sensor Systems*. Oct. 2011, pp. 192–201. DOI: `10.1109/MASS.2011.26`.

[55] Y. Li, D. Jin, Z. Wang, L. Zeng, and S. Chen. “Coding or Not: Optimal Mobile Data Offloading in Opportunistic Vehicular Networks.” In: *IEEE Transactions on Intelligent Transportation Systems* 15.1 (Feb. 2014), pp. 318–333. ISSN: 1524-9050. DOI: `10.1109/TITS.2013.2281104`.

[56] F. Dressler, H. Hartenstein, O. Altintas, and O. K. Tonguz. “Inter-vehicle communication: Quo vadis.” In: *IEEE Communications Magazine* 52.6 (June 2014), pp. 170–177. ISSN: 0163-6804. DOI: `10.1109/MCOM.2014.6829960`.

[57] M. Lee, J. Song, J. Jeong, and T. Kwon. “DOVE: Data Offloading through Spatio-Temporal Rendezvous in Vehicular Networks.” In: *2015 24th International Conference on Computer Communication and Networks (ICCCN)*. Aug. 2015, pp. 1–8. DOI: `10.1109/ICCCN.2015.7288400`.

[58] European Telecommunications Standards Institute. “Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Definitions.” In: ETSI TR 102 638, V1.1.1 (2009).

[59] European Telecommunications Standards Institute. “Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service.” In: ETSI TS 102 637-2, V1.1.1 (2010).

[60] European Telecommunications Standards Institute. "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service." In: ETSI EN 302 637-3, V1.2.1 (2014).

[61] European Telecommunications Standards Institute. "Intelligent Transport Systems (ITS); Users and applications requirements; Part 1: Facility layer structure, functional requirements and specifications." In: ETSI TS 102 894-1, V1.1.1 (2013).

[62] Alassane Samba, Yann Busnel, Alberto Blanc, Philippe Dooze, and Gwendal Simon. "Predicting File Downloading Time in Cellular Network: Large-Scale Analysis of Machine Learning Approaches." In: *Computer Networks* (Sept. 8, 2018). ISSN: 1389-1286. DOI: `10.1016/j.comnet.2018.09.002`. URL: `http://www.sciencedirect.com/science/article/pii/S1389128618308545` (Last accessed on August 1, 2019).

[63] Nicola Bui, Foivos Michelinakis, and Joerg Widmer. "A model for throughput prediction for mobile users." In: *European Wireless 2014; 20th European Wireless Conference; Proceedings of*. VDE, 2014, pp. 1–6. URL: `http://ieeexplore.ieee.org/abstract/document/6843210/` (Last accessed on August 1, 2019).

[64] Yan Liu and Jack YB Lee. "An Empirical Study of Throughput Prediction in Mobile Data Networks." In: *Global Communications Conference (GLOBECOM), 2015 IEEE*. IEEE, 2015, pp. 1–6. URL: `http://ieeexplore.ieee.org/abstract/document/7417858/` (Last accessed on August 1, 2019).

[65] Nicola Bui, Foivos Michelinakis, and Joerg Widmer. "Fine-grained LTE Radio Link Estimation for Mobile Phones." In: (2017). URL: `http://eprints.networks.imdea.org/1562/` (Last accessed on August 1, 2019).

[66] Christoph Ide, Michael Nick, Dennis Kaulbars, and Christian Wietfeld. "Forecasting Cellular Connectivity for Cyber-Physical Systems: A Machine Learning Approach." In: *Machine Learning for Cyber Physical Systems*. Ed. by Oliver Niggemann and Jürgen Beyerer. DOI: 10.1007/978-3-662-48838-6_3. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 15–22. ISBN: 978-3-662-48836-2. URL: `http://link.springer.com/10.1007/978-3-662-48838-6_3` (Last accessed on August 1, 2019).

[67] Christian Wietfeld, Christoph Ide, and Bjoern Dusza. "Resource Efficient Mobile Communications for Crowd-Sensing." In: ACM Press, 2014, pp. 1–6. ISBN: 978-1-4503-2730-5. DOI: `10.1145/2593069.2596686`. URL: `http://dl.acm.org/citation.cfm?doid=2593069.2596686` (Last accessed on August 1, 2019).

[68] Ruofan Jin. "Enhancing Upper-level Performance from Below: Performance Measurement and Optimization in LTE Networks." In: (2015). URL: `http://digitalcommons.uconn.edu/dissertations/981/?utm_source=digitalcommons.uconn.edu%2Fdissertations%2F981&utm_medium=PDF&utm_campaign=PDFCoverPages` (Last accessed on August 1, 2019).

[69] Alassane Samba, Yann Busnel, Alberto Blanc, Philippe Dooze, and Gwendal Simon. "Instantaneous Throughput Prediction in Cellular Networks: Which Information Is Needed?" In: *2017 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. IEEE, 2017. URL: `https://hal.archives-ouvertes.fr/hal-01535696/` (Last accessed on August 1, 2019).

[70] Alassane Samba, Yann Busnel, Alberto Blanc, Philippe Dooze, and Gwendal Simon. "Throughput Prediction in Cellular Networks: Experiments and Preliminary Results." In: *CoRes 2016*. 2016. URL: `https://hal.archives-ouvertes.fr/hal-01311158/` (Last accessed on August 1, 2019).

[71] Qiang Xu, Sanjeev Mehrotra, Zhuoqing Mao, and Jin Li. "PROTEUS: network performance forecast for real-time, interactive mobile applications." In: *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*. ACM, 2013, pp. 347–360. URL: `http://dl.acm.org/citation.cfm?id=2464453` (Last accessed on August 1, 2019).

[72] Jun Yao, Salil S. Kanhere, and Mahbub Hassan. "An empirical study of bandwidth predictability in mobile computing." In: *Proceedings of the third ACM international workshop on Wireless network testbeds, experimental evaluation and characterization*. ACM, 2008, pp. 11–18. URL: `http://dl.acm.org/citation.cfm?id=1410081` (Last accessed on August 1, 2019).

[73] Feng Lu, Hao Du, Ankur Jain, Geoffrey M. Voelker, Alex C. Snoeren, and Andreas Terzis. "CQIC: Revisiting Cross-Layer Congestion Control for Cellular Networks." In: ACM Press, 2015, pp. 45–50. ISBN: 978-1-4503-3391-7. DOI: `10.1145/2699343.2699345`. URL: `http://dl.acm.org/citation.cfm?doid=2699343.2699345` (Last accessed on August 1, 2019).

[74] Robert Margolies, Ashwin Sridharan, Vaneet Aggarwal, Rittwik Jana, N. K. Shankaranarayanan, Vinay A. Vaishampayan, and Gil Zussman. "Exploiting Mobility in Proportional Fair Cellular Scheduling: Measurements and Algorithms." In: *IEEE/ACM Trans. Netw.* 24.1 (Feb. 2016), pp. 355–367. ISSN: 1063-6692. DOI: `10.1109/TNET.2014.2362928`. URL: `Piscataway,%20NJ,%20USA`.

[75] Lutz Kelch, Tobias Pogel, Lars Wolf, and Angela Sasse. "CQI Maps for Optimized Data Distribution." In: *Vehicular Technology Conference (VTC Fall), 2013 IEEE 78th*. IEEE, 2013, pp. 1–5. (Last accessed on August 1, 2019).

[76] Tobias Pögel and Lars Wolf. "Optimization of vehicular applications and communication properties with Connectivity Maps." In: *Local Computer Networks Conference Workshops (LCN Workshops), 2015 IEEE 40th*. IEEE, 2015, pp. 870–877. URL: `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7365940` (Last accessed on August 1, 2019).

[77] Tobias Pögel and Lars Wolf. "Prediction of 3G network characteristics for adaptive vehicular Connectivity Maps (Poster)." In: *Vehicular Networking Conference (VNC), 2012 IEEE*. IEEE, 2012, pp. 121–128. URL: `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6407420` (Last accessed on August 1, 2019).

[78] Theodoros Kamakaris and Jeffrey V. Nickerson. “Connectivity maps: Measurements and applications.” In: *System Sciences, 2005. HICSS’05. Proceedings of the 38th Annual Hawaii International Conference on*. IEEE, 2005, pp. 307–307. URL: `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1385874` (Last accessed on August 1, 2019).

[79] Robert Nagel and Stefan Morscher. *Connectivity Prediction in Mobile Vehicular Environments Backed By Digital Maps*. INTECH Open Access Publisher, 2011. URL: `http://cdn.intechopen.com/pdfs-wm/14257.pdf` (Last accessed on August 1, 2019).

[80] Benjamin Sliwa, Thomas Liebig, Robert Falkenberg, Johannes Pillmann, and Christian Wietfeld. “Machine learning based context-predictive car-to-cloud communication using multi-layer connectivity maps for upcoming 5G networks.” In: *arXiv:1805.06603 [cs]* (May 17, 2018). arXiv: `1805.06603`. URL: `http://arxiv.org/abs/1805.06603` (Last accessed on August 1, 2019).

[81] Edwin Bastiaensen et al. “ActMAP: real-time map updates for advanced in-vehicle applications.” In: *Proceedings, 10th World Congress on ITS, Madrid*. 2003.

[82] Kyoung-Wook Min, Kyoung-Hwan An, Ju-Wan Kim, and Sung-Il Jin. “The mobile spatial DBMS for the partial map air update in the navigation.” In: *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*. IEEE, 2008, pp. 476–481. URL: `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4732538` (Last accessed on August 1, 2019).

[83] SeungGwan Lee and Sungwon Lee. “Map Generation and Updating Technologies based on Network and Cloud Computing: A Survey.” In: *International Journal of Multimedia and Ubiquitous Engineering*. Vol. 8. 4. 2013, pp. 107–114. URL: `http://www.sersc.org/journals/IJMUE/vol8_no4_2013/11.pdf` (Last accessed on August 1, 2019).

[84] Nicola Bui and Joerg Widmer. “OWL: a Reliable Online Watcher for LTE Control Channel Measurements.” In: *ACM All Things Cellular (MobiCom Workshop)*. New York, USA, Nov. 2016.

[85] Dietbert Guetter. “Ausbreitung elektromagnetischer Wellen.” In: (2016), p. 55.

[86] F. Beritelli, G. Capizzi, G. Lo Sciuto, C. Napoli, and F. Scaglione. “Rainfall Estimation Based on the Intensity of the Received Signal in a LTE/4G Mobile Terminal by Using a Probabilistic Neural Network.” In: *IEEE Access* 6 (2018), pp. 30865–30873. ISSN: 2169-3536. DOI: `10.1109/ACCESS.2018.2839699`.

[87] Junxian Huang, Feng Qian, Alexandre Gerber, Z. Morley Mao, Subhabrata Sen, and Oliver Spatscheck. “A close examination of performance and power characteristics of 4G LTE networks.” In: *Proceedings of the 10th international conference on Mobile systems, applications, and services*. ACM, 2012, pp. 225–238. URL: `http://dl.acm.org/citation.cfm?id=2307658` (Last accessed on August 1, 2019).

[88] Florian Jomrich, Aakash Sharma, Tobias Rückelt, Doreen Böhnstedt, and Ralf Steinmetz. "Intelligent Offloading Distribution of High Definition Street Maps for Highly Automated Vehicles." In: *Smart Cities, Green Technologies, and Intelligent Transport Systems 6th International Conference, SMARTGREENS 2017, and Third International Conference, VEHITS 2017, Porto, Portugal, April 22-24, 2017, Revised Selected Papers*. Ed. by Brian Donnellan et al. Springer, Aug. 2017. ISBN: 978-3-030-02906-7.

[89] Florian Jomrich, Daniel Bischoff, Steffen Knapp, Tobias Meuser, Björn Richerzhagen, and Ralf Steinmetz. "Lane Accurate Detection of Map Changes based on Low Cost Smartphone Data." English. In: *In Proceedings of the 5th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS)*. Vol. 5. May 2019, p. 12.

[90] Tobias Rückelt. "Connecting Vehicles to the Internet - Strategic Data Transmission for Mobile Nodes using Heterogeneous Wireless Networks." PhD thesis. Technische Universität, Nov. 2017. URL: `http://tuprints.ulb.tu-darmstadt.de/6913/`.

[91] Mathias Schneider, Josef Schmidt, Alfred Höß, and Florian Jomrich. "Vehicle to Server Communication for Highly Automated Driving in the Ko-HAF Project." In: *Application-Oriented Higher Education Research (AOHER)* (Mar. 2019), p. 13. ISSN: 2096-2045.

[92] Florian Jomrich, Josef Schmid, Steffen Knapp, Alfred Höß, Ralf Steinmetz, and Björn Schuller. "Analysing communication requirements for crowd sourced backend generation of HD Maps used in automated driving." In: *Proceedings of the 2018 IEEE Vehicular Networking Conference (VNC)*. Dec. 2018, p. 8.

[93] Florian Jomrich, Markus Grau, Tobias Meuser, The An Binh Nguyen, Doreen Böhnstedt, and Ralf Steinmetz. "ICCOMQS - Intelligent measuring framework to ensure reliable communication for highly automated vehicles." In: *Proceedings of 2017 IEEE Vehicular Networking Conference (VNC)*. IEEE, Nov. 2017, pp. 311–318. ISBN: 978-1-5386-0985-9.

[94] Florian Jomrich, Alexander Herzberger, Tobias Meuser, Björn Richerzhagen, Ralf Steinmetz, and Cornelius Wille. "Cellular Bandwidth Prediction for Highly Automated Driving - Evaluation of Machine Learning Approaches based on Real-World Data." In: *Proceedings of the 4th International Conference on Vehicle Technology and Intelligent Transport Systems 2018*. Ed. by Markus Helfert and Oleg Gusikhin. 4. SCITEPRESS, Mar. 2018, pp. 121–131. ISBN: 978-989-758-293-6.

[95] Florian Jomrich, Florian Fischer, Steffen Knapp, Tobias Meuser, Björn Richerzhagen, and Ralf Steinmetz. "Enhanced Cellular Bandwidth Prediction for Highly Automated Driving." In: *Smart Cities, Green Technologies, and Intelligent Transport Systems 7th International Conference, SMARTGREENS 2018, and Fourth International Conference, VEHITS 2018, Funchal, Madeira, Portugal, March 16-18, 2018, Revised Selected Papers*. Ed. by Brian Donnellan et al. Springer, Dec. 2019. ISBN: 978-3-030-26633-2.

[96] Florian Jomrich, Tushar Wankhede, Tobias Rückelt, Daniel Burgsthaler, Doreen Böhnstedt, and Ralf Steinmetz. "Demo: Rapid Cellular Network Simulation Framework for Automotive Scenarios (RACE Framework)." English. In: *Proceedings of: International Conference on Networked Systems (NetSys 2017)*. IEEE, Mar. 2017, pp. 1–2. ISBN: 978-1-5090-4395-8.

[97] Florian Jomrich, Tobias Rückelt, Doreen Böhnstedt, and Ralf Steinmetz. "An Efficient Heat-Map-Based Wireless Communication Simulation Model for Omnet++." English. In: *Proceedings of the AmE 2017 - Automotive meets Electronics*. Vol. 8. Berlin · Offenbach, Bismarckstraße 33, 10625 Berlin: VDE Verlag GmbH, Mar. 2017, pp. 65–70. ISBN: 978-3-8007-4369-8.

[98] Hoang T. Dinh, Chonho Lee, Dusit Niyato, and Ping Wang. "A survey of mobile cloud computing: architecture, applications, and approaches." In: *Wireless Communications and Mobile Computing* 13.18 (Dec. 25, 2013), pp. 1587–1611. ISSN: 1530-8669. DOI: `10.1002/wcm.1203`. URL: `https://onlinelibrary.wiley.com/doi/full/10.1002/wcm.1203` (Last accessed on August 1, 2019).

[99] Yun Chao Hu, Dario Sabella, Nurit Sprecher, Valerie Young, and Milan Patel. "Mobile Edge Computing A key technology towards 5G." In: *ETSI White Paper No. 11* (First Edition 2015), p. 16.

[100] SAE International. "Levels of Driving Automation - SAE International Standard J3016." In: (2014).

[101] Daniel Burgstahler, Christoph Peusens, Doreen Boehnstedt, and Ralf Steinmetz. "Horizon.KOM: A First Step Towards an Open Vehicular Horizon Provider:" in: *Proceedings of the International Conference on Vehicle Technology and Intelligent Transport Systems*. International Conference on Vehicle Technology and Intelligent Transport Systems. Rome, Italy: SCITEPRESS - Science, 2016, pp. 79–84. ISBN: 978-989-758-185-4. DOI: `10.5220/0005799700790084`. URL: `http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0005799700790084` (Last accessed on August 1, 2019).

[102] Christian Ress, Aria Etemad, Detlef Kuck, and Julián Requejo. "Electronic Horizon - Providing Digital Map Data for ADAS Applications:" in: *Proceedings of the 2nd International Workshop on Intelligent Vehicle Control Systems*. The Second International Workshop on Intelligent Vehicle Control Systems. Funchal - Madeira, Portugal: SciTePress - Science, 2008, pp. 40–49. ISBN: 978-989-8111-34-0. DOI: `10.5220/0001508000400049`. URL: `http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0001508000400049` (Last accessed on August 1, 2019).

[103] Leica Geosystems AG. "GPS Basics." In: (2000). URL: `http://webarchiv.ethz.ch/geometh-data/downloads/GPSBasics_de.pdf` (Last accessed on August 1, 2019).

[104] Iping Supriana Suwardi, Dody Dharma, Dicky Prima Satya, and Dessi Puji Lestari. "Geohash index based spatial data model for corporate." In: *Electrical Engineering and Informatics (ICEEI), 2015 International Conference on*. IEEE, 2015, pp. 478–483. URL: `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7352548` (Last accessed on August 1, 2019).

[105] Daniel Bischoff, Harald Berninger, Steffen Knapp, Tobias Meuser, Björn Richerzhagen, Lars Häring, and Andreas Czylwik. "Safety-relevant V2X beaconing in realistic and scalable heterogeneous radio propagation fading channels." In: *Proceedings of the International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS)*. SCITEPRESS, May 2019. URL: `ftp://ftp.kom.tu-darmstadt.de/papers/BBK+19.pdf`.

[106] Michael Behrisch, Laura Bieker, Jakob Erdmann, and Daniel Krajzewicz. "SUMO–simulation of urban mobility: an overview." In: *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*. ThinkMind. 2011.

[107] Florian Hagenauer, Falko Dressler, and Christoph Sommer. "Poster: A simulator for heterogeneous vehicular networks." In: *2014 IEEE Vehicular Networking Conference (VNC)*. IEEE, 2014, pp. 185–186. URL: `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7013339` (Last accessed on August 1, 2019).

[108] B. Sliwa, J. Pillmann, F. Eckermann, L. Habel, M. Schreckenberg, and C. Wietfeld. "Lightweight joint simulation of vehicular mobility and communication with LIMoSim." In: *2017 IEEE Vehicular Networking Conference (VNC)*. Nov. 2017, pp. 81–88. DOI: `10.1109/VNC.2017.8275600`.

[109] Lara Codecá, Raphaël Frank, Sébastien Faye, and Thomas Engel. "Luxembourg SUMO Traffic (LuST) Scenario: Traffic Demand Evaluation." In: *IEEE Intelligent Transportation Systems Magazine* 9.2 (2017), pp. 52–63.

[110] Lara Codeca and Jérôme Härri. "Towards multimodal mobility simulation of C-ITS: The Monaco SUMO traffic scenario." In: *VNC 2017, IEEE Vehicular Networking Conference, November 27-29, 2017, Torino, Italy*. Nov. 2017. DOI: `http://dx.doi.org/10.1109/VNC.2017.8275627`. URL: `http://www.eurecom.fr/publication/5350`.

[111] Leo Kent. *Autonomous cars can only understand the real world through a map*. 2015. URL: `https://360.here.com/2015/04/16/autonomous-cars-can-understand-real-world-map//` (Last accessed on August 1, 2019).

[112] Y. L. Morgan. "Notes on DSRC amp; WAVE Standards Suite: Its Architecture, Design, and Characteristics." In: *IEEE Communications Surveys Tutorials* 12.4 (Apr. 2010), pp. 504–518. ISSN: 1553-877X. DOI: `10.1109/SURV.2010.033010.00024`.

[113] Alessio Filippi, Kees Moerman, Vincent Martinez, Andrew Turley, Onn Haran, and Ron Toledano. "IEEE802.11p ahead of LTE-V2V for safety applications." In: (2017), p. 19.

[114] Lan Lin. *ETSI G5 technology: the European approach*. 2013. URL: `http://www.drive-c2x.eu/tl%5C_files/publications/3rd%5C%20Test%5C%20Site%5C%20Event%5C%20TSS/1%5C%20DRIVE%5C%20C2X%5C%203rd%5C%20Test%5C%20site%5C%20event%5C_Lan%5C%20Lin%5C_Technology%5C_20130613.pdf`.

[115] Michelle Weigele. *Standards: WAVE/DSRC/802.11p*. 2008. URL: `http://www.cvt-project.ir/Admin/Files/eventAttachments/109.pdf`.

[116] Yi Shi. "LTE-V: a cellular-assisted V2X communication technology." In: *ITU Workshop*. 2015.

[117] 5G Automotive Association Director General Dino Flore. "5G V2X, The automotive use-case for 5G." In: (2017).

[118] S. Chen, J. Hu, Y. Shi, Y. Peng, J. Fang, R. Zhao, and L. Zhao. "Vehicle-to-Everything (v2x) Services Supported by LTE-Based Systems and 5G." In: *IEEE Communications Standards Magazine* 1.2 (2017), pp. 70–76. ISSN: 2471-2825. DOI: `10.1109/MCOMSTD.2017.1700015`.

[119] Cen B. Liu, Bahareh Sadeghi, and Edward W. Knightly. "Enabling vehicular visible light communication (V2LC) networks." In: *Proceedings of the Eighth ACM international workshop on Vehicular inter-networking - VANET '11*. the Eighth ACM international workshop. Las Vegas, Nevada, USA: ACM Press, 2011, p. 41. ISBN: 978-1-4503-0869-4. DOI: `10.1145/2030698.2030705`. URL: `http://dl.acm.org/citation.cfm?doid=2030698.2030705` (Last accessed on August 1, 2019).

[120] P. Ji, H. Tsai, C. Wang, and F. Liu. "Vehicular Visible Light Communications with LED Taillight and Rolling Shutter Camera." In: *2014 IEEE 79th Vehicular Technology Conference (VTC Spring)*. 2014 IEEE 79th Vehicular Technology Conference (VTC Spring). May 2014, pp. 1–6. DOI: `10.1109/VTCSpring.2014.7023142`.

[121] Agon Memedi, Claas Tebruegge, Julien Jahneke, and Falko Dressler. "Impact of Vehicle Type and Headlight Characteristics on Vehicular VLC Performance." In: *2018 IEEE Vehicular Networking Conference (VNC)*. 2018 IEEE Vehicular Networking Conference (VNC). Taipei, Taiwan: IEEE, Dec. 2018, pp. 1–8. ISBN: 978-1-5386-9428-2. DOI: `10.1109/VNC.2018.8628444`. URL: `https://ieeexplore.ieee.org/document/8628444/` (Last accessed on August 1, 2019).

[122] E. Schoch, F. Kargl, and M. Weber. "Communication patterns in VANETs." In: *IEEE Communications Magazine* 46.11 (Nov. 2008), pp. 119–125. ISSN: 0163-6804. DOI: `10.1109/MCOM.2008.4689254`.

[123] F. Malandrino, C. Casetti, C. F. Chiasserini, and M. Fiore. "Offloading cellular networks through ITS content download." In: *2012 9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*. June 2012, pp. 263–271. DOI: `10.1109/SECON.2012.6275786`.

[124] Fusang Zhang, Hai Liu, Yiu-Wing Leung, Xiaowen Chu, and Beihong Jin. "CBS: Community-Based Bus System as Routing Backbone for Vehicular Ad Hoc Networks." In: *IEEE Transactions on Mobile Computing* 16.8 (Aug. 2017), pp. 2132–2146. ISSN: 1536-1233. DOI: `10.1109/TMC.2016.2613869`. URL: `http://ieeexplore.ieee.org/document/7577744/` (Last accessed on August 1, 2019).

[125] Aws Saad, Tareq Rahem Abdalrazak, Ammar Jameel Hussein, and Amer Mohammed Abdullah. "Vehicular Ad Hoc Networks: Growth and Survey for Three Layers." In: *International Journal of Electrical and Computer Engineering* 7.1 (2017), p. 271. URL: `http://search.proquest.com/openview/62b1f563b7b6b468dfeee25370f72898/1?pq-origsite=gscholar%5C&cbl=1686344` (Last accessed on August 1, 2019).

[126] Aws Saad, Tareq Rahem Abdalrazak, Ammar Jameel Hussein, and Amer Mohammed Abdullah. "Vehicular Ad Hoc Networks: Growth and Survey for Three Layers." In: *International Journal of Electrical and Computer Engineering* 7.1 (2017), p. 271.

[127] Liming Liu, Tao Wu, Yuqiang Fang, Tingbo Hu, and Jinze Song. "A smart map representation for autonomous vehicle navigation." In: *Fuzzy Systems and Knowledge Discovery (FSKD), 2015 12th International Conference on*. IEEE, 2015, pp. 2308–2313. URL: `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7382313` (Last accessed on August 1, 2019).

[128] Avdhut Joshi and Michael R. James. "Generation of Accurate Lane-Level Maps from Coarse Prior Maps and Lidar." In: *IEEE Intelligent Transportation Systems Magazine* 7.1 (2015), pp. 19–29. ISSN: 1939-1390. DOI: `10.1109/MITS.2014.2364081`. URL: `http://ieeexplore.ieee.org/document/7014398/` (Last accessed on August 1, 2019).

[129] Dietmar Rabel. *How HERE HD Live Map perpetually heals itself*. 2017. URL: `https://360.here.com/a-map-that-perpetually-heals-itself-the-hd-live-map` (Last accessed on August 1, 2019).

[130] Greg Miller. *Autonomous Cars Will Require a Totally New Kind of Map | WIRED*. 2014. URL: `http://www.wired.com/2014/12/nokia-here-autonomous-car-maps/` (Last accessed on August 1, 2019).

[131] J. Pauls, T. Strauss, C. Hasberg, M. Lauer, and C. Stiller. "Can We Trust Our Maps? An Evaluation of Road Changes and a Dataset for Map Validation." In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. Nov. 2018, pp. 2639–2644. DOI: `10.1109/ITSC.2018.8569249`.

[132] "New traffic data sources – An overview." In: *BITRE/IPA joint workshop: 'Exploring New Sources of Traffic Data'*. BITRE, 2014.

[133] K. Massow, B. Kwella, N. Pfeifer, F. Häusler, J. Pontow, I. Radusch, J. Hipp, F. Dölitzscher, and M. Haueis. "Deriving HD maps for highly automated driving from vehicular probe data." In: *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*. IEEE, 2016, pp. 1745–1752. URL: `http:`

//ieeexplore.ieee.org/abstract/document/7795794/ (Last accessed on August 1, 2019).

[134] Brian Niehöfer and et al. "GPS Community Map Generation for Enhanced Routing Methods Based on Trace-Collection by Mobile Phones." In: IEEE, 2009, pp. 156–161. DOI: 10.1109/SPACOMM.2009.31. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5194603.

[135] ERIACM Uduwaragoda, A. S. Perera, and S. A. D. Dias. "Generating lane level road data from vehicle trajectories using kernel density estimation." In: *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. IEEE, 2013, pp. 384–391.

[136] Dr. Lukas Klejnowski, Dr. Markus Rausch, Dr. Artur Quiring, and Dr. Holger Mielenz. "Towards HAD-map-change Detection on Backend Server." In: *In proceedings of 4th In-ternational VDI Conference 2017 – Automated Driving*. Robert Bosch GmbH, 2017, pp. 26–31.

[137] René Brüntrup and et al. "Incremental map generation with GPS traces." In: *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE*. IEEE, 2005, pp. 574–579. (Last accessed on August 1, 2019).

[138] Lili Cao and John Krumm. "From GPS traces to a routable road map." In: *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*. ACM, 2009, pp. 3–12. URL: http://dl.acm.org/citation.cfm?id=1653776 (Last accessed on August 1, 2019).

[139] Jonathan J. Davies and et al. "Scalable, distributed, real-time map generation." In: *IEEE Pervasive Computing* 5.4 (2006), pp. 47–54. URL: http://ieeexplore.ieee.org/abstract/document/1717365/ (Last accessed on August 1, 2019).

[140] K.D. McDonlad and C. Hegarty. *Postmodernization GPS Performance Capabilites*. Proc. IAIN World Congress and the ION 56th Ann. Meeting. Inst. of Navigation, 2000. 242-249.

[141] R. Prasad and M.Ruggieri. *Applied Satellite Navigation Using GPS, GALILEO , and Augmentation Systems*. Artech House, 2005.

[142] Mahmuda Ahmed and et al. "A comparison and evaluation of map construction algorithms using vehicle tracking data." In: *GeoInformatica* 19.3 (2015), pp. 601–632.

[143] David Betaille and Rafael Toledo-Moreo. "Creating Enhanced Maps for Lane-Level Vehicle Navigation." In: *IEEE Transactions on Intelligent Transportation Systems* 11.4 (Dec. 2010), pp. 786–798. ISSN: 1524-9050, 1558-0016. DOI: 10.1109/TITS.2010.2050689. URL: http://ieeexplore.ieee.org/document/5499153/ (Last accessed on August 1, 2019).

[144] Yihua Chen and John Krumm. "Probabilistic modeling of traffic lanes from GPS traces." In: *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2010, pp. 81–88. URL: http://dl.acm.org/citation.cfm?id=1869805 (Last accessed on August 1, 2019).

[145] Nobuyoshi Sato, Tsuyoshi Takayama, and Yoshitoshi Murata. "Estimating the Number of Lanes on Rapid Road Map Survey System Using GPS Trajectories as Collective Intelligence." In: IEEE, Sept. 2012, pp. 82–88. ISBN: 978-1-4673-2331-4. DOI: `10.1109/NBiS.2012.49`.

[146] Zhidan Liu and et al. "A Participatory Urban Traffic Monitoring System: The Power of Bus Riders." In: *IEEE Transactions on Intelligent Transportation Systems* (2017), pp. 1–14. ISSN: 1524-9050, 1558-0016. DOI: `10.1109/TITS.2017.2650215`. URL: `http://ieeexplore.ieee.org/document/7837684/` (Last accessed on August 1, 2019).

[147] E. Kaffashi, M. T. Shoorabi, and S. H. Bojnourdi. "Coverage optimization in wireless sensor networks." In: *2014 4^{th} International Conference on Computer and Knowledge Engineering (ICCKE)*. Oct. 2014, pp. 322–327. DOI: `10.1109/ICCKE.2014.6993344`.

[148] M. Argany, M. A. Mostafavi, and C. Gagné. "Context-Aware Local Optimization of Sensor Network Deployment." In: *Journal of Sensor and Actuator Networks* 4.3 (2015), pp. 160–188.

[149] Jorge Cortés, Sonia Martínez, Timur Karatas, and Francesco Bullo. "Coverage Control for Mobile Sensing Networks." In: *ICRA*. IEEE, 2002, pp. 1327–1332. ISBN: 0-7803-7273-5. URL: `http://dblp.uni-trier.de/db/conf/icra/icra2002.html#CortesMKB02`.

[150] M. Zhong and C. G. Cassandras. "Distributed Coverage Control and Data Collection With Mobile Sensor Networks." In: *IEEE Transactions on Automatic Control* 56.10 (Oct. 2011), pp. 2445–2455. ISSN: 0018-9286. DOI: `10.1109/TAC.2011.2163860`.

[151] Tobias Pögel, Jan Lübbe, and Lars Wolf. "Passive client-based bandwidth and latency measurements in cellular networks." In: *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*. IEEE, 2012, pp. 37–42. URL: `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6193516` (Last accessed on August 1, 2019).

[152] Foivos Michelinakis, Nicola Bui, Guido Fioravantti, Joerg Widmer, Fabian Kaup, and David Hausheer. "Lightweight capacity measurements for mobile networks." In: *Computer Communications* 84 (2016), pp. 73–83. URL: `http://www.sciencedirect.com/science/article/pii/S014036641630024X` (Last accessed on August 1, 2019).

[153] Yin Xu, Zixiao Wang, Wai Kay Leong, and Ben Leong. "An end-to-end measurement study of modern cellular data networks." In: *International Conference on Passive and Active Network Measurement*. Springer, 2014, pp. 34–45. URL: `http://link.springer.com/chapter/10.1007/978-3-319-04918-2_4` (Last accessed on August 1, 2019).

[154] Pavlos Papageorge, Justin McCann, and Michael Hicks. "Passive aggressive measurement with MGRP LESEN." In: *ACM SIGCOMM Computer Communication Review*. Vol. 39. ACM, 2009, pp. 279–290. URL: `http://dl.acm.org/citation.cfm?id=1592601` (Last accessed on August 1, 2019).

[155] C. Yue, R. Jin, K. Suh, Y. Qin, B. Wang, and W. Wei. "LinkForecast: Cellular Link Bandwidth Prediction in LTE Networks." In: *IEEE Transactions on Mobile Computing* (2017), pp. 1–1. ISSN: 1536-1233. DOI: `10.1109/TMC.2017.2756937`.

[156] Swarun Kumar, Ezzeldin Hamed, Dina Katabi, and Li Erran Li. "LTE Radio Analytics Made Easy and Accessible." In: *Proceedings of the 6th Annual Workshop on Wireless of the Students, by the Students, for the Students*. S3 '14. Maui, Hawaii, USA: ACM, 2014, pp. 29–30. ISBN: 978-1-4503-3073-2. DOI: `10.1145/2645884.2645891`. URL: `New%20York,%20NY,%20USA`.

[157] Robert Falkenberg, Christoph Ide, and Christian Wietfeld. "Client-Based Control Channel Analysis for Connectivity Estimation in LTE Networks." In: *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)* (Sept. 2016), pp. 1–6. DOI: `10.1109/VTCFall.2016.7880932`. arXiv: `1701.03304`. URL: `http://arxiv.org/abs/1701.03304` (Last accessed on August 1, 2019).

[158] Robert Falkenberg, Karsten Heimann, and Christian Wietfeld. "Discover Your Competition in LTE: Client-Based Passive Data Rate Prediction by Machine Learning." In: *GLOBECOM 2017 - 2017 IEEE Global Communications Conference* (Dec. 2017), pp. 1–7. DOI: `10.1109/GLOCOM.2017.8254567`. arXiv: `1711.06820`. URL: `http://arxiv.org/abs/1711.06820` (Last accessed on August 1, 2019).

[159] András Varga et al. "The OMNeT++ discrete event simulation system." In: *Proceedings of the European simulation multiconference (ESM'2001)*. Vol. 9. sn, 2001, p. 65. URL: `https://labo4g.enstb.fr/twiki/pub/Simulator/SimulatorReferences/esm2001-meth48.pdf` (Last accessed on August 1, 2019).

[160] Gustavo Carneiro. "NS-3: Network simulator 3." In: *UTM Lab Meeting April*. Vol. 20. 2010.

[161] Robert Protzmann, Bjoern Schuenemann, and Ilja Radusch. "Simulation of Convergent Networks for Intelligent Transport Systems with VSimRTI: High Mobile Wireless Nodes." In: Apr. 2017, pp. 1–28. ISBN: 9781848218536. DOI: `10.1002/9781119407447.ch1`.

[162] Rimon Barr, Zygmunt J Haas, and R Van Renesse. "Jist/swans." In: *Wireless Networks Laboratory, Cornell University. http://jist. ece. cornell. edu* (2005).

[163] Christoph Sommer, Reinhard German, and Falko Dressler. "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis." In: *IEEE Transactions on Mobile Computing* 10.1 (Jan. 2011), pp. 3–15. DOI: `10.1109/TMC.2010.133`.

[164] Nicola Baldo. *The ns-3 LTE module by the LENA project*. 2011. URL: `http://www2.nsnam.org/tutorials/tutorials/consortium13/lte-tutorial.pdf` (Last accessed on August 1, 2019).

[165] Antonio Virdis, Giovanni Stea, and Giovanni Nardini. "Simulating LTE/LTE-Advanced Networks with SimuLTE." In: *Springer Advances in Intelligent Systems and Computing* 402.January (2015). DOI: `10.1007/978-3-319-26470-7`.

[166] Mohammad S. Obaidat, Janusz Kacprzyk, Tuncer Ören, and Joaquim Filipe, eds. *Simulation and modeling methodologies, technologies and applications: International Conference, SIMULTECH 2014 Vienna, Austria, August 28-30, 2014 Revised Selected Papers*. Advances in intelligent systems and computing Volume 402. OCLC: 950868128. Cham Heidelberg New York Dordrecht London: Springer International Publishing, 2015. 352 pp. ISBN: 978-3-319-26469-1.

[167] R. Riebl, H. Günther, C. Facchi, and L. Wolf. "Artery: Extending Veins for VANET applications." In: *2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*. June 2015, pp. 450–456. DOI: `10.1109/MTITS.2015.7223293`.

[168] *GitHub - ibr-cm/artery-lte: ArteryLTE is a holistic IVC simulation framework capable of simulating different communication technologies (incl. ETSI ITS-G5 and LTE)*. URL: `https://github.com/ibr-cm/artery-lte` (Last accessed on August 1, 2019).

[169] Scott Biddlestone, Keith Redmill, Radovan Miucic, and Ümit Ozguner. "An integrated 802.11 p WAVE DSRC and vehicle traffic simulator with experimentally validated urban (LOS and NLOS) propagation models." In: *IEEE transactions on intelligent transportation systems* 13.4 (2012), pp. 1792–1802.

[170] Florian Jomrich, Aakash Sharma, Tobias Rückelt, Daniel Burgstahler, and Doreen Böhnstedt. "Dynamic Map Update Protocol for Highly Automated Driving Vehicles." English. In: *Proceedings of the 3rd International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2017)*. Ed. by Oleg Gusikhin, Markus Helfert, and António Pascoal. Vol. 3. Full Paper. SCITEPRESS – Science and Technology Publications, Lda., Apr. 2017, pp. 68–78. ISBN: 978-989-758-242-4.

[171] VIRES. *OpenDRIVE - managing the road ahead*. 2011. URL: `http://www.opendrive.org/%20docs/VIRES%5C_ODR%5C_OCRG.pdf` (Last accessed on August 1, 2019).

[172] NDS Navigation Data Standard e.V. *NDS Open Lane Model Press Release*. 2016. URL: `http://%20www.nds-association.org/wp-content/uploads/20160914-PR-E.pdf` (Last accessed on August 1, 2019).

[173] Philipp Bender, Julius Ziegler, and Christoph Stiller. "Lanelets: Efficient map representation for autonomous driving." In: *2014 IEEE Intelligent Vehicles Symposium Proceedings*. IEEE, 2014, pp. 420–425. URL: `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6856487` (Last accessed on August 1, 2019).

[174] Mordechai Haklay and Patrick Weber. "Openstreetmap: User-generated street maps." In: *IEEE Pervasive Computing*. Vol. 7. 4. 2008, pp. 12–18. URL: `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4653466` (Last accessed on August 1, 2019).

[175] Björn Richerzhagen, Dominik Stingl, Julius Rückert, and Ralf Steinmetz. "Simonstrator: Simulation and Prototyping Platform for Distributed Mobile Applications." In: *Proc. 8th International Conference on Simulation Tools and Techniques (SIMUTOOLS)*. ACM. Aug. 2015, pp. 99–108.

[176] K. Graffi. "PeerfactSim.KOM: A P2P system simulator — Experiences and lessons learned." In: *2011 IEEE International Conference on Peer-to-Peer Computing*. Aug. 2011, pp. 154–155. DOI: 10.1109/P2P.2011.6038673.

[177] Pino Bonetti. *HERE introduces HD Live Map to show the path to highly automated driving - HERE 360*. URL: http://360.here.com/2016/01/05/here-introduces-hd-live-map-to-show-the-path-to-highly-automated-driving/%20[online%20accessed%202019-05-15] (Last accessed on August 1, 2019).

[178] TomTom. *HD Map - highly accuarte border to border model of the road*. 2017. URL: http://%20download.tomtom.com/open/banners/HD-Map-Product-Info-Sheet-improved-1.pdf%20[online%20accessesd%202019-05-15] (Last accessed on August 1, 2019).

[179] HERE. *Vehicle Sensor Data Cloud Ingestion Interface Specification (v2.0.2)*. 2015. URL: https://lts.cms.here.com/static-cloud-content/Company_Site/2015_06/Vehicle_Sensor_Data_Cloud_Ingestion_Interface_Specification.pdf.

[180] ERTICO. *ERTICO coordinated platform releases first on vehicle-to-cloud data standard*. July 2018. URL: http://erticonetwork.com/ertico-coordinated-platform-releases-first-on-vehicle-to-cloud-data-standard/.

[181] Raj Kishore Prasad, Deepali Sale, and Mrinai M Dhanvijay. "Survey of MobileCommunication Systems." In: 6.2 (2019), p. 7.

[182] C. Johnson. *Long Term Evolution in Bullets, 2nd Edition*. Createspace Independent Pub, 2012. ISBN: 9781452834641. URL: https://books.google.de/books?id=-suOSQAACAAJ.

[183] Christopher Haslett. *Essentials of Radio Wave Propagation*. The Cambridge Wireless Essentials Series. Cambridge University Press, 2007. DOI: 10.1017/CBO9780511536762.

[184] *Entscheidung der Präsidentenkammer der Bundesnetzagentur für Elektrizität, Gas, Telekommunikation, Post und Eisenbahnen vom 28. Januar 2015 zur Anordnung und Wahl des Verfahrens sowie über die Festlegungen und Regeln im Einzelnen (Vergaberegeln) und über die Festlegungen und Regelungen für die Durchführung des Verfahrens (Auktionsregeln) zur Vergabe von Frequenzen in den Bereichen 700 MHz, 900 MHz, 1800 MHz sowie weiterer Frequenzen im Bereich 1452 – 1492 MHz für den drahtlosen Netzzugang zum Angebot von Telekommunikationsdiensten;* Deutschland: Bundesnetzagentur, Jan. 28, 2015. URL: https://www.bundesnetzagentur.de/SharedDocs/Downloads/DE/Sachgebiete/Telekommunikation/Unternehmen_Institutionen/Frequenzen/OffentlicheNetze/Mobilfunk/%20DrahtloserNetzzugang/Projekt2016/EntscheidungProjekt2016_pdf.pdf?__blob=publicationFile&v=1 (Last accessed on August 1, 2019).

[185] Bundesnetzagentur. *Konsultationsentwurf einer Entscheidung der Präsidentenkammer der Bundesnetzagentur für Elektrizität, Gas, Telekommunikation, Post und Eisenbahnen über die Fest-legungen und Regeln im Einzelnen (Vergaberegeln) und über die Festlegungen und Regelungen für die Durchführung des Verfahrens (Auktionsregeln) zur Vergabe von Frequenzen in den Bereichen 2 GHz und 3,6 GHz.* BK1-17/001. Germany, Sept. 24, 2018.

[186] Bundesnetzagentur. *Frequenzplan gemäß § 54 TKG über die Aufteilung des Frequenzbereichs von 0 kHz bis 3000 GHz auf die Frequenznutzungen sowie über die Festlegungen für diese Frequenznutzungen.* Mar. 1, 2018. URL: `https://www.bundesnetzagentur.de/SharedDocs/Downloads/DE/Sachgebiete/%20Telekommunikation/Unternehmen_Institutionen/Frequenzen/%20Frequenzplan.pdf?__blob=publicationFile&v=11` (Last accessed on August 1, 2019).

[187] Richard Möller, Peter Jonsson, Stephen Carson, Jasmeet Singh Sethi, Mats Arvedson, Ritva Svenningsson, Per Lindberg, Kati Öhman, Patrik Hedlund, and Veronica Gully. “Ericsson Mobility Report November 2017.” In: (2017), p. 32.

[188] Cisco. *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016–2021 - White Paper*. Feb. 7, 2017. URL: `https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.pdf` (Last accessed on August 1, 2019).

[189] Cisco. *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2017–2022*. Feb. 1, 2019. URL: `https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-738429.pdf` (Last accessed on August 1, 2019).

[190] BAST. *Automatische Zählstellen 2016, Dauerzählstelle: Frankfurt-Niederrad*. 2016. URL: `https://www.bast.de/BASt_2017/DE/Verkehrstechnik/Fachthemen/v2-verkehrszaehlung/Daten/2016_1/Jawe2016.html?nn=1819490&cms_detail=6923&cms_map=0`.

[191] W. Afrić and S. Z. Pilinsky. “Multipath fading and LTE downlink cell size calculation.” In: *Proceedings ELMAR-2013*. Sept. 2013, pp. 259–262.

[192] Sandra Neubauer. *LKW-Geschwindigkeit auf der Autobahn: Diese Limits gelten*. May 20, 2019. URL: `https://www.bussgeldkatalog.org/lkw-geschwindigkeit-autobahn/` (Last accessed on August 1, 2019).

[193] Kai Biermann, Paul Blickle, Andreas Loos, and Sascha Venohr. *Wo Deutschland rast*. Apr. 10, 2019. URL: `https://www.zeit.de/mobilitaet/2019-02/autobahnen-geschwindigkeit-tempo-schnelligkeit-raser-verkehr` (Last accessed on August 1, 2019).

[194] Adam Langley, Alistair Riddoch, Alyssa Wilk, Antonio Vicente, Charles Krasic, Dan Zhang, Fan Yang, Fedor Kouranov, Ian Swett, Janardhan Iyengar, Jeff Bailey, Jeremy Dorfman, Jim Roskind, Joanna Kulik, Patrik Westin, Raman Tenneti, Robbie Shade, Ryan Hamilton, Victor Vasiliev, Wan-Teh Chang, and

Zhongyi Shi. “The QUIC Transport Protocol: Design and Internet-Scale Deployment.” In: *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. SIGCOMM ’17. Los Angeles, CA, USA: ACM, 2017, pp. 183–196. ISBN: 978-1-4503-4653-5. DOI: `10.1145/3098822.3098842`. URL: `New%20York,%20NY,%20USA`.

[195] J. Yao, S. S. Kanhere, and M. Hassan. “Mobile Broadband Performance Measured from High-Speed Regional Trains.” In: *2011 IEEE Vehicular Technology Conference (VTC Fall)*. Sept. 2011, pp. 1–5. DOI: `10.1109/VETECF.2011.6092874`.

[196] Yuanjie Li, Chunyi Peng, Zengwen Yuan, Jiayao Li, Haotian Deng, and Tao Wang. “Mobileinsight: Extracting and Analyzing Cellular Network Information on Smartphones.” In: *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. MobiCom ’16. New York City, New York: ACM, 2016, pp. 202–215. ISBN: 978-1-4503-4226-1. DOI: `10.1145/2973750.2973751`. URL: `New%20York,%20NY,%20USA`.

[197] Tobias Rueckelt, Daniel Burgstahler, Doreen Böhnstedt, and Ralf Steinmetz. “MoVeNet : Mobility Management for Vehicular Networking.” In: *Proceedings of the ACM International Symposium on Mobility Management and Wireless Access (MobiWAC)*. 2016. ISBN: 1234567245. DOI: `10.475/123`.

[198] Josep Colom Ikuno, Martin Wrulich, and Markus Rupp. “System Level Simulation of LTE Networks.” In: *Proceedings of the IEEE Vehicular Technology Conference (VTC)*. 2010. ISBN: 978-1-4244-2518-1. DOI: `10.1109/VETECS.2010.5494007`.

[199] Lauren Isaac. *Will We Ever See Level 5 (Fully Automated) Driving?* 2018. URL: `https://drivingtowardsdriverless.com/2018/06/06/will-we-ever-see-level-5-fully-automated-driving/` (Last accessed on August 1, 2019).

[200] Todd Litman. “Implications for Transport Planning.” In: *Autonomous Vehicle Implementation Predictions* (2019), p. 39.

[201] Karl Hemlin and Frida Persson. *Remote Control Operation of Autonomous Cars Over Cellular Network Using PlayStation Controller*. 2019.

[202] W. Shi and S. Dustdar. “The Promise of Edge Computing.” In: *Computer* 49.5 (May 2016), pp. 78–81. ISSN: 0018-9162. DOI: `10.1109/MC.2016.145`.

[203] Z. Pi and F. Khan. “An introduction to millimeter-wave mobile broadband systems.” In: *IEEE Communications Magazine* 49.6 (June 2011), pp. 101–107. ISSN: 0163-6804. DOI: `10.1109/MCOM.2011.5783993`.

[204] W. Roh, J. Seol, J. Park, B. Lee, J. Lee, Y. Kim, J. Cho, K. Cheun, and F. Aryanfar. “Millimeter-wave beamforming as an enabling technology for 5G cellular communications: theoretical feasibility and prototype results.” In: *IEEE Communications Magazine* 52.2 (Feb. 2014), pp. 106–113. ISSN: 0163-6804. DOI: `10.1109/MCOM.2014.6736750`.

[205] D. Phan-Huy, M. Sternad, and T. Svensson. “Making 5G Adaptive Antennas Work for Very Fast Moving Vehicles.” In: *IEEE Intelligent Transportation Systems Magazine* 7.2 (2015), pp. 71–84. ISSN: 1939-1390. DOI: `10.1109/MITS.2015.2408151`.

[206] 3GPP. "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation." In: TS 36.211-f60 (2019).

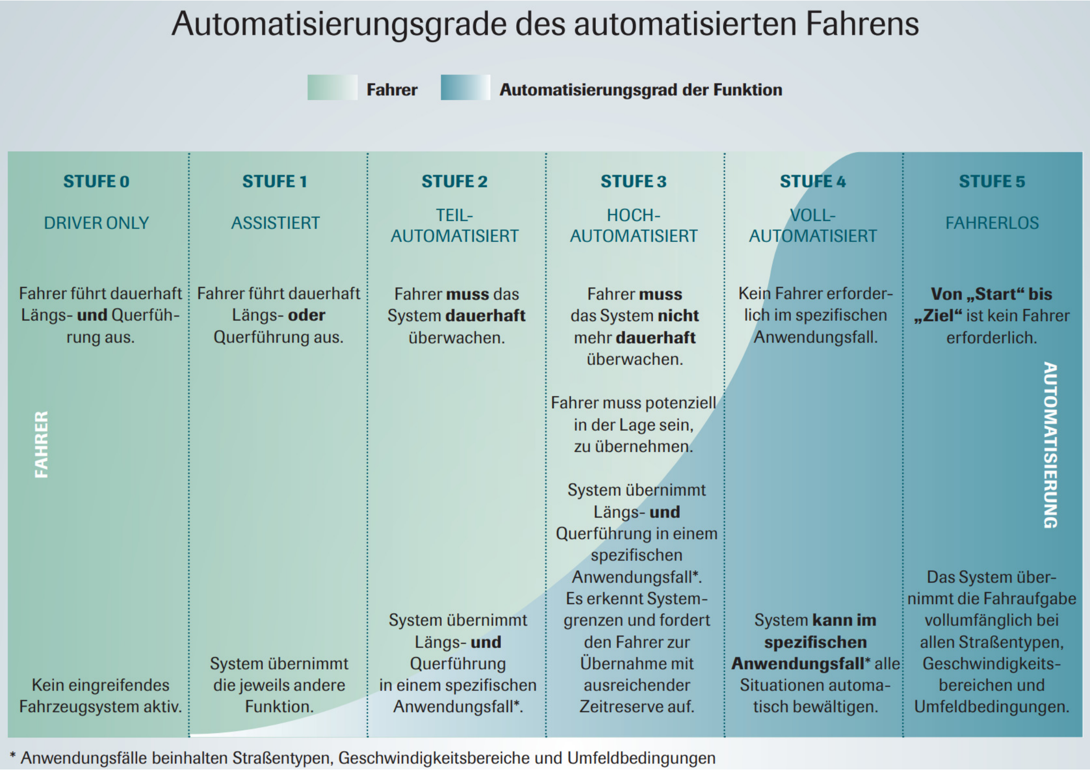
All web pages cited in this work have been checked in August 2019. However, due to the dynamic nature of the World Wide Web, their long-term availability cannot be guaranteed.

A

APPENDIX

A.1 DEFINITION OF AUTOMATION LEVELS

The following two Figures 70 and 71 illustrate the definition of the various automation levels specified through the German Association of the Automotive Industry (Verband der Automobilindustrie - VDA) and through the Society of Automotive Engineers - SAE. Both definitions have to be carefully considered separately from each other when speaking of the different automation levels, as they do not associate the same degree of automation with the same specified level. In the presented work the definition of the VDA is considered as the reference. Furthermore we assume a level 3 - highly automated (hochautomatisiert) vehicle when using the term of a self-driving vehicle.



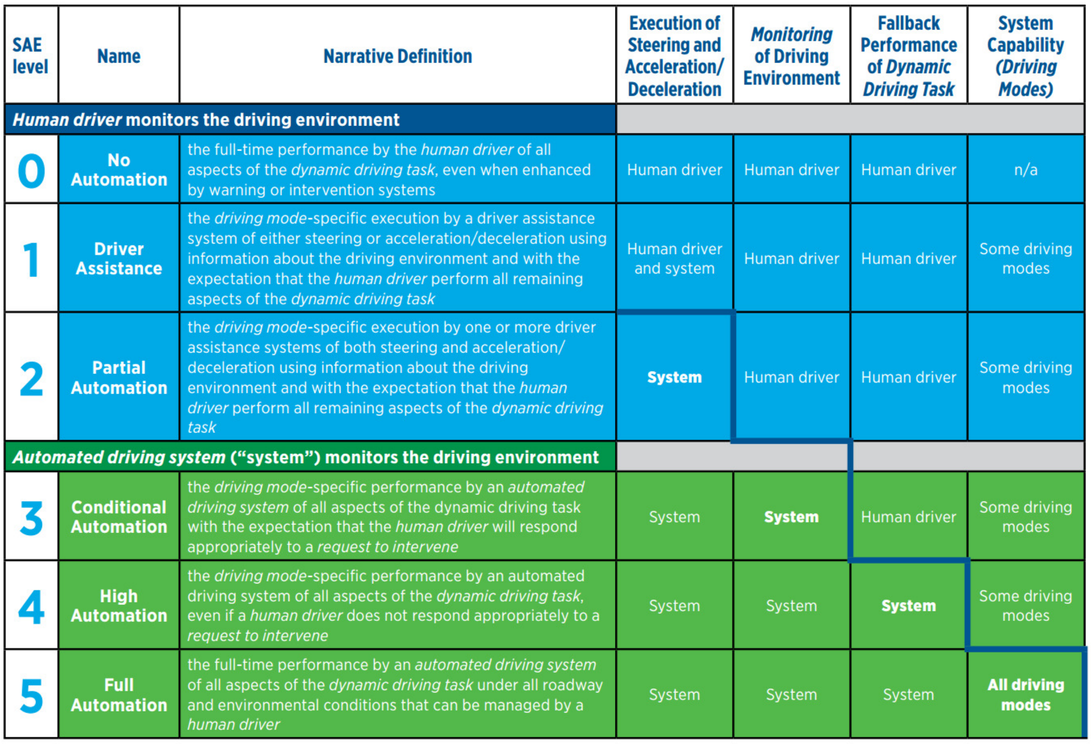
Automatisierungsgrade des automatisierten Fahrens

🟩 Fahrer 🟦 Automatisierungsgrad der Funktion

STUFE 0	STUFE 1	STUFE 2	STUFE 3	STUFE 4	STUFE 5
DRIVER ONLY	ASSISTIERT	TEIL-AUTOMATISIERT	HOCH-AUTOMATISIERT	VOLL-AUTOMATISIERT	FAHRERLOS
Fahrer führt dauerhaft Längs- **und** Querführung aus.	Fahrer führt dauerhaft Längs- **oder** Querführung aus.	Fahrer **muss** das System **dauerhaft** überwachen.	Fahrer **muss** das System **nicht** mehr **dauerhaft** überwachen. Fahrer muss potenziell in der Lage sein, zu übernehmen.	Kein Fahrer erforderlich im spezifischen Anwendungsfall.	**Von „Start“ bis „Ziel“** ist kein Fahrer erforderlich.
Kein eingreifendes Fahrzeugsystem aktiv.	System übernimmt die jeweils andere Funktion.	System übernimmt Längs- **und** Querführung in einem spezifischen Anwendungsfall*.	System übernimmt Längs- **und** Querführung in einem spezifischen Anwendungsfall*. Es erkennt Systemgrenzen und fordert den Fahrer zur Übernahme mit ausreichender Zeitreserve auf.	System **kann im spezifischen Anwendungsfall*** alle Situationen automatisch bewältigen.	Das System übernimmt die Fahraufgabe vollumfänglich bei allen Straßentypen, Geschwindigkeitsbereichen und Umfeldbedingungen.

* Anwendungsfälle beinhalten Straßentypen, Geschwindigkeitsbereiche und Umfeldbedingungen

Figure 70: Overview of the Levels of Driving Automation for On-Road Vehicles as defined by the German Association of the Automotive industry (Verband der Automobilindustrie - VDA) (Source: page 15 in [2]). ©VDA



SAE level	Name	Narrative Definition	Execution of Steering and Acceleration/ Deceleration	*Monitoring* of Driving Environment	Fallback Performance of *Dynamic Driving Task*	System Capability *(Driving Modes)*
Human driver **monitors the driving environment**						
0	**No Automation**	the full-time performance by the *human driver* of all aspects of the *dynamic driving task*, even when enhanced by warning or intervention systems	Human driver	Human driver	Human driver	n/a
1	**Driver Assistance**	the *driving mode*-specific execution by a driver assistance system of either steering or acceleration/deceleration using information about the driving environment and with the expectation that the *human driver* perform all remaining aspects of the *dynamic driving task*	Human driver and system	Human driver	Human driver	Some driving modes
2	**Partial Automation**	the *driving mode*-specific execution by one or more driver assistance systems of both steering and acceleration/ deceleration using information about the driving environment and with the expectation that the *human driver* perform all remaining aspects of the *dynamic driving task*	**System**	Human driver	Human driver	Some driving modes
Automated driving system **("system") monitors the driving environment**						
3	**Conditional Automation**	the *driving mode*-specific performance by an *automated driving system* of all aspects of the dynamic driving task with the expectation that the *human driver* will respond appropriately to a *request to intervene*	System	**System**	Human driver	Some driving modes
4	**High Automation**	the *driving mode*-specific performance by an automated driving system of all aspects of the *dynamic driving task*, even if a *human driver* does not respond appropriately to a *request to intervene*	System	System	**System**	Some driving modes
5	**Full Automation**	the full-time performance by an *automated driving system* of all aspects of the *dynamic driving task* under all roadway and environmental conditions that can be managed by a *human driver*	System	System	System	**All driving modes**

Copyright © 2014 SAE International. The summary table may be freely copied and distributed provided SAE International and J3016 are acknowledged as the source and must be reproduced AS-IS.

Figure 71: Overview of the Levels of Driving Automation for On-Road Vehicles as defined by the SAE International (former Society of Automotive Engineers) [100]. ©SAE

A.2 PROTOTYPICAL IMPLEMENTATION OF AN OPERATIONAL HD MAP - EXAMPLE OF THE KO-HAF PROJECT

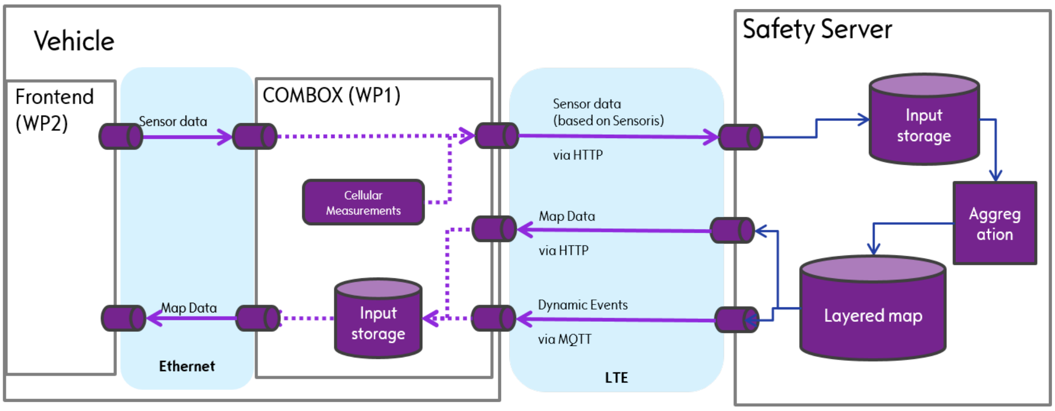


Figure 72: Overview about the prototypical realization of a functional HD Map with data exchange, as conducted in Ko-HAF. ©Ko-HAF[37]

To give a better insight into the general working principle of an operational HD Map we describe its prototypical realization in the German government founded research project Ko-HAF (Cooperative Highly Automated Driving) in the following. Based on this realization our own research contributions have been conducted. Several major participants of the German automotive industry took part in the Ko-HAF project [91, 92], including OEMs (Audi, BMW, Opel) and Tier 1 suppliers (Bosch and Continental). All of them rely on different sensor equipment within their cars to realize the driving functionality. Consequently a common communication interface between them and the central Safety Server, which hosts and maintains the HD Map data, had to be developed as one of Ko-HAF's major contributions. The final realization of this communication and processing work flow is described by Figure 72. The two communication entities, on the one hand the self-driving vehicle, on the other hand the central Safety Server, are connected through a cellular interface with each other. To realize the required data exchange between the different vehicles and the server an identical communication unit (COM-Box), as shown in Figure 73, was installed in the fleet of twelve participating cars. To reliable exchange the sensor and map data between the vehicles frontend and the COM-Box inside the car, a common interface was defined using Ethernet connections. The data packets containing the sensor data of the vehicle, e.g. the detection of lane markings and traffic signs (see e.g. Fig 39), were further enriched by additional sensor data provided by the COM-Box regarding the currently available quality of the cellular network transmission (see Section 6.1 for further details). To ensure the availability of the HD Map data required for the following few track kilometers, even if the cellular connection might become unstable, the COM-Box possesses an internal map cache in which always the map data for a few road kilometers ahead is stored. Should the car not be able to download further

---

[37] https://ko-haf.de/fileadmin/user_upload/media/abschlusspraesentation/13_Ko-HAF_Creation-and-Deployment-of-HD-Map-Data.pdf (Last accessed on August 1, 2019)

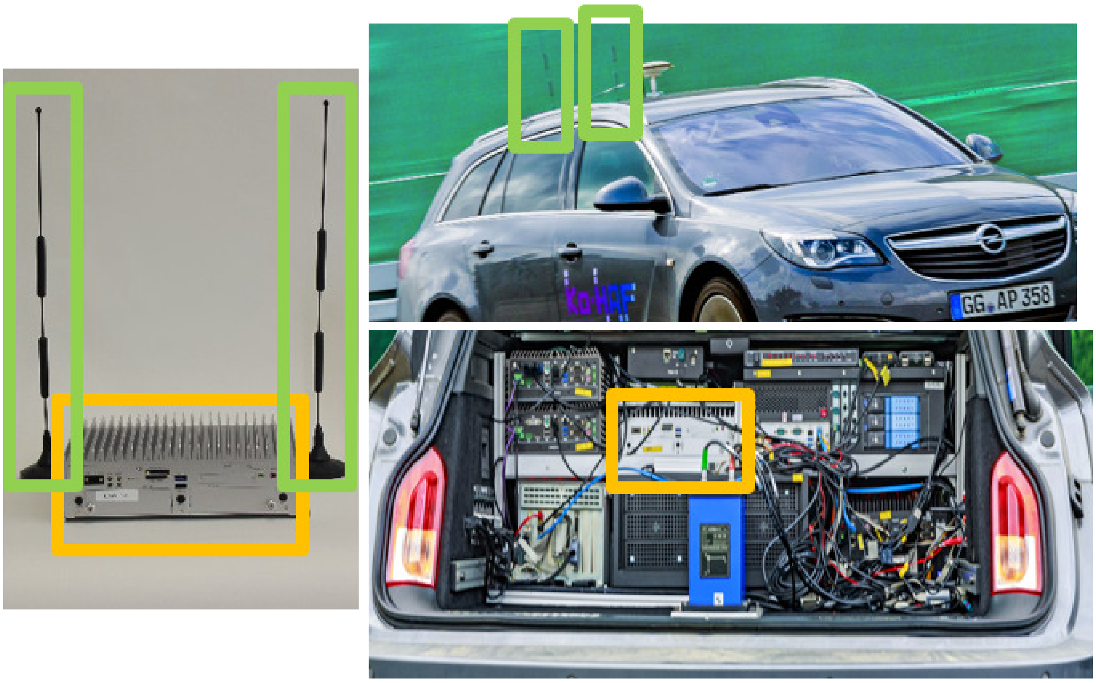


Figure 73: The common communication box installed in all twelve vehicles, that participated in the Ko-HAF Project. The box provided communication functionalities via all cellular network technologies (4G, 3G and 2G), that were deployed during the duration of the Ko-HAF project (September 2015 - November 2018). ©Ko-HAF, Opel Automobile GmbH[39]

map data this cached part of the map ensures enough time for a safe handover back to the human driver. The sensor data itself represented through the SENSORIS car-to-cloud universal data standard [179] is serialized before its transmission using Google Protocol Buffers[38]. The map tiles provided from the Safety Server to the vehicles are transferred using Google Protocol Buffers, too. That way we ensure an overall very data-efficient transmission process. All the incoming sensor data provided by the self-driving vehicles and other 3rd parties are then stored first inside an input buffer in the Safety Server. If a reliable amount of data was collected that way, a downstream aggregation process was executed to update the layered map. The required amount of such sensor data to maintain the HD Map itself is still part of ongoing research. We conducted own research contributions in that domain to quickly identify map changes to ensure the safety of the self-driving car (see Chapter 5). Furthermore we identified the areas of poor network quality (see Chapter 6) to provide the self-driving vehicle with additional knowledge regarding the cellular infrastructure to schedule its personal data transmissions [66, 67, 80, 90] accordingly.

---

[38] https://developers.google.com/protocol-buffers/ (Last accessed on August 1, 2019)

[39] https://ko-haf.de/fileadmin/user_upload/media/abschlusspraesentation/14_Ko-HAF_Continuous-Updating-of-Backend-HD-Map-Data.pdf (Last accessed on August 1, 2019)

A.3 KEY PERFORMANCE INDICATORS OF THE CELLULAR NETWORK

In the following we give a detailed explanation of all cellular key performance indicators (KPI), which are relevant for our personal work described in Chapter 6. The listed key performance indicators are directly related to the LTE cellular network [206]. Most of them can either be applied directly on other wireless network technologies (e.g. cellular network technologies of second, third or fifth generation) or mapped on similar parameters.

A.3.1 *Reference Signal Received Power - RSRP*

The Reference Signal Received Power - RSRP is defined as the linear average of the power amount of the resource elements (RE), which carry the cell-specific reference signals accordingly to the currently used frequency bandwidth.

The common range for measurable values of the RSRP is between -140 dBm and -50 dBm. The higher the value the better.

A.3.2 *Received Signal Strength Indicaton - RSSI*

The Received Signal Strength Indicaton (RSSI) is a measurement for the power present in all of the received radio signal, including forms of noise. Common RSSI values of LTE range between -90 dBm and -120 dBm. The higher the value the better.

A.3.3 *Reference Signal Received Quality - RSRQ*

The RSRQ is a value derived from the RSRP and RSSI value as defined in Formula 9, where N is the number of resource blocks related to the currently used frequency bandwidth. It commonly ranges between -3 dB and -20 dB. The higher the value the better.

$$\mathrm{RSRQ}[W] = \mathrm{NxRSRP}[W]/\mathrm{RSSI}[W] \tag{9}$$

In combination with the RSRP value the RSRQ provides and indicator to identify the best possible position and orientation of the mobile end device (UE) towards its currently serving celltower.

A.3.4 *Channel Quality Indicator - CQI*

The Channel Quality Indiator (CQI) is estimated by the mobile end device (UE) to inform its currently serving cell tower about the currently experienced signal quality. Based on the CQI and its current load the celltower then selects a suitable Modulation and Coding Scheme for the UE.

A.3.5 *Carrier frequency as EARFCN*

The E-UTRA Absolute Radio Frequency Channel Number (EARFCN) is an indicator for the currently used transmission frequencies in the uplink and the downlink. The EARFCN numbers also can be mapped on one of the defined LTE bands. During our cellular measurements as described in Section 6.1 we experienced transmission frequencies of the three German providers in the LTE bands 1 (2100 MHz), 3 (1800 MHz), 7 (2600 MHz) and 20 (800MHz)[40].

A.3.6 *Carrier Aggregation*

Carrier Aggregation is an LTE-Advanced feature first introduced in LTE Release 10[41]. The technique allows the network operators to combine different separate physical frequency bands into one single "virtual band". This allows to reach higher bandwidths for the transmission of data, as the operator's available spreaded spectrum in the different frequency bands (e.g. 10 MHz in LTE Band 20 [800 MHz] and 20 MHz in LTE Band 7 [2600 MHz] combined to a virtual 30 MHz band) would allow individually. It allows the operators to utilize their owned spectrum more efficiently, as most of them don't have large, coherent frequency bands at their disposal. The increase in available bandwidth directly correlates with a higher achievable bandwidth.

A.3.7 *Multiple Input Multiple Output - MIMO*

Through the usage of several distinct transmission and receiving antennas the mobile end device (e.g a smartphone) and the cell tower, to which it is connected, can exchange data over different physical data streams in parallel on the same transmission frequency. On the receiving side, the various streams are differentiated through the received signal strength of each individual signal at the receiving antennas. This concept is called Multiple Input Multiple Output or short MIMO and is not only common for cellular transmission techniques, but also other wireless transmission techniques such as WLAN. MIMO can significantly improve the realizable coding rate of the signal and thus the overall achieved transmission speed.

A.3.8 *Modulation and Coding - Scheme*

The achievable data throughput in LTE cellular networks correlates with the received signal strength at the site of the mobile end device. The smartphone for example informs its serving cellular tower about its currently experienced signal quality via the so called Channel Quality Indicator (CQI). Based on the received CQI the cell tower then decides which Modulation and Coding Scheme (MCS) it should use to transmit data to and from the end device to ensure a robust connection under the conditions

---

[40] https://www.spectrummonitoring.com/frequencies/#Germany (Last accessed on August 1, 2019)

[41] https://www.3gpp.org/specifications/releases (Last accessed on August 1, 2019)

of the experienced signal quality. Depending on the LTE device's category and its currently received signal strength modern day LTE modems (e.g. Qualcomm's Snapdragon X24 LTE modem[42] as representative of a device in the LTE Category 20) are able to upload or download data via four different modulation methods (QPSK, 16 QAM, 64 QAM and 256 QAM). In correlation with the selected Modulation an according Coding Scheme is selected, which enables through error correction techniques a fast and robust data transmission.

A.3.9 *Resource Blocks and Transport Block Size*

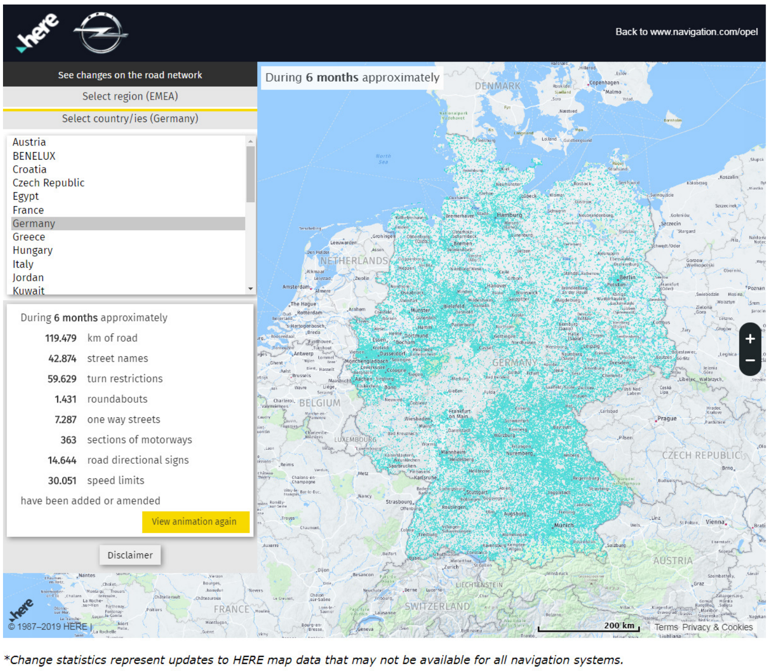
In the LTE protocol a Resource Block is considered as the smallest unit of data that can be transmitted. Therefore the available frequency spectrum of the transmission channel (between 1.4 MHz and 20 MHz for a single carrier) are assigned different numbers of Resource Blocks. 10 MHz of bandwidth for example represent 50 Resource Blocks and 20 MHz resemble 100 Resource Blocks. Each cell tower schedules the transmission of data every 1 ms. Within this time period it reassigns it's available Resource Blocks between all its connected clients. In correlation with the currently selected Modulation and Coding Scheme (Sec. A.3.8) one Resource Block thus can transmit a certain amount of data within this 1 ms time period. This amount of data then is represented by the Transport Block Size (TBS). The Transport Block Size thus directly correlates with the achievable throughput of the cellular data connection.

---

[42] https://www.qualcomm.com/products/snapdragon-x24-lte-modem (Last accessed on August 1, 2019)

A.4 MAP CHANGES IN DIGITAL NAVIGATION MAP DATA OF HERE

The following graphics 74 and 75 illustrate the amount of road infrastructure changes in Here's navigation map data of Germany over different periods of time (6, 12 and 18 months).



During **6 months** approximately

119.479	km of road
42.874	street names
59.629	turn restrictions
1.431	roundabouts
7.287	one way streets
363	sections of motorways
14.644	road directional signs
30.051	speed limits

have been added or amended

**Change statistics represent updates to HERE map data that may not be available for all navigation systems.*

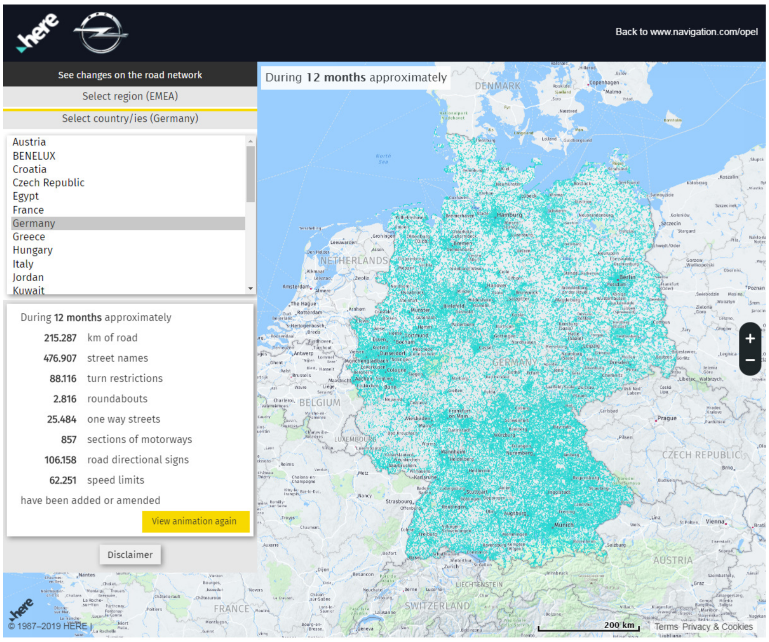
(a) after 6 months

Figure 74: Amount of changes in the standard navigational map data provided by Here for navigation systems of Opel's production vehicles for the area of Germany over various time periods (map data as of 14.01.2019). ©Here and Opel Automobile GmbH[43]

---

[43] http://mapchanges.navigation.com/?app=opel (Last accessed on August 1, 2019)

[44] http://mapchanges.navigation.com/?app=opel (Last accessed on August 1, 2019)



During **12 months** approximately

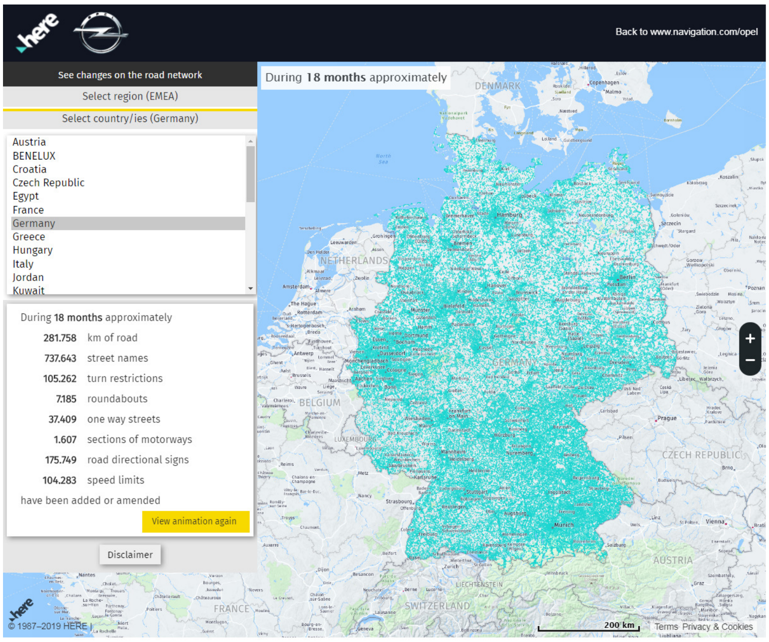
During **12 months** approximately

215.287	km of road
476.907	street names
88.116	turn restrictions
2.816	roundabouts
25.484	one way streets
857	sections of motorways
106.158	road directional signs
62.251	speed limits

have been added or amended

**Change statistics represent updates to HERE map data that may not be available for all navigation systems.*

(a) after 12 months



During **18 months** approximately

During **18 months** approximately

281.758	km of road
737.643	street names
105.262	turn restrictions
7.185	roundabouts
37.409	one way streets
1.607	sections of motorways
175.749	road directional signs
104.283	speed limits

have been added or amended

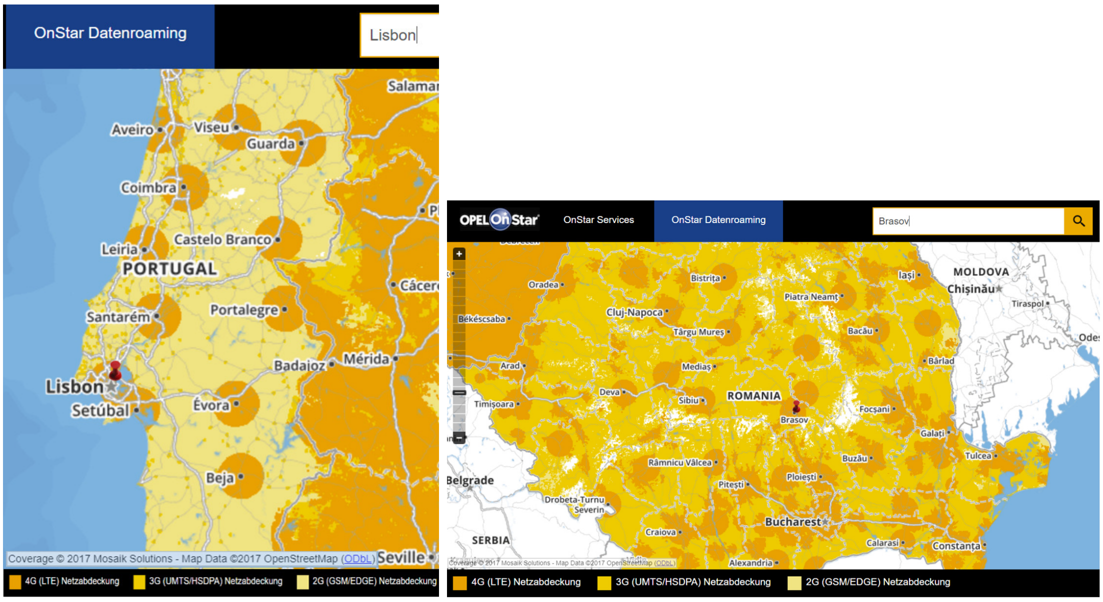
**Change statistics represent updates to HERE map data that may not be available for all navigation systems.*

(b) after 18 months

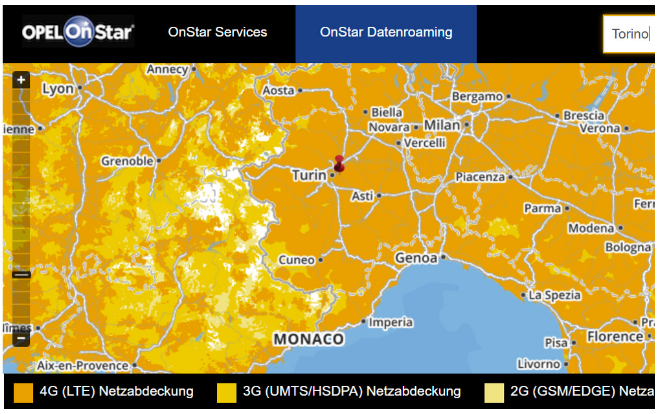
Figure 75: Continued amount of changes in the standard navigational map data provided by Here for navigation systems of Opel's production vehicles for the area of Germany over various time periods (map data as of 14.01.2019). ©Here and Opel Automobile GmbH[44]

A.5 NETWORK COVERAGE MAPS OF CELLULAR PROVIDERS

In the following network coverage maps for the Opel OnStar Cellular Service in different regions in Europe are presented. Clearly visible are the assumed circular regions of service coverage, which likely do not resemble the actual availability of the service.



(a) Portugal (b) Romania



(c) Turin - North Italy

Figure 76: Network coverage maps of the Opel OnStar Cellular Service in different locations in Europe. ©Opel Automobile GmbH[45], Mosaik Solutions, OpenStreetMap Contributors

---

[45] https://www.opel.at/onstar/onstar-verfuegbarkeit.html (Last accessed on August 1, 2019)

A.6 DETAILED LISTING OF THE EVALUATION RESULTS OF THE HD-WMAP EXTENSION OF THE DYNAMIC MAP UPDATE PROTOCOL

Table 18 summarizes the detailed evaluation results of our HD-Wmap extension. The impact of buses behaving as data beacons, which already have the complete map data for their route, in comparison to them behaving like normal vehicles, is only marginal.

configured scenario	ad hoc sharing quota mean [%]	ad hoc sharing quota median [%]	necessary cellular transmission quota mean [%]	necessary cellular transmission quota median [%]
buses as normal vehicles	43.6	35.1	56.4	64.9
buses as data beacons	44.3	35.3	55.7	64.7

Table 18: Achieved ad hoc sharing quotas of the HD-Wmap extension, when considering buses either as normal vehicles or as data beacons with an extended initial map data storage.

A.7 RIGHT LANE EVALUATION RESULTS OF THE LANE COURSE DETECTION ALGORITHM

In the following the evaluation results for the right lane data of our proposed lane course detection algorithm (see Chapter 5) are presented.

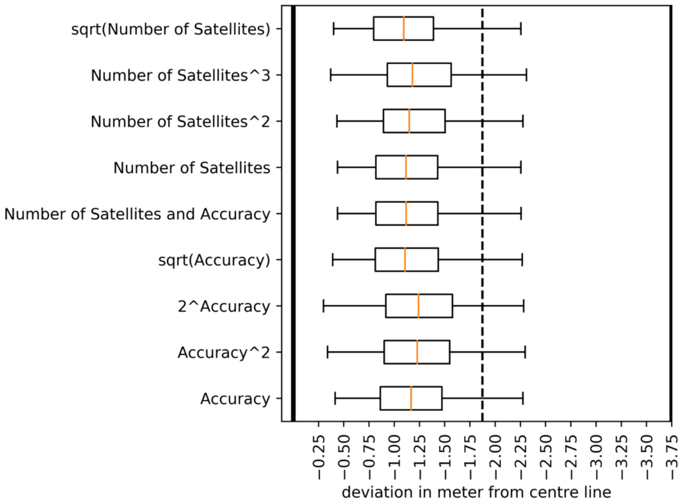


Figure 77: Comparison of different weighting functions (right lane data) [89].

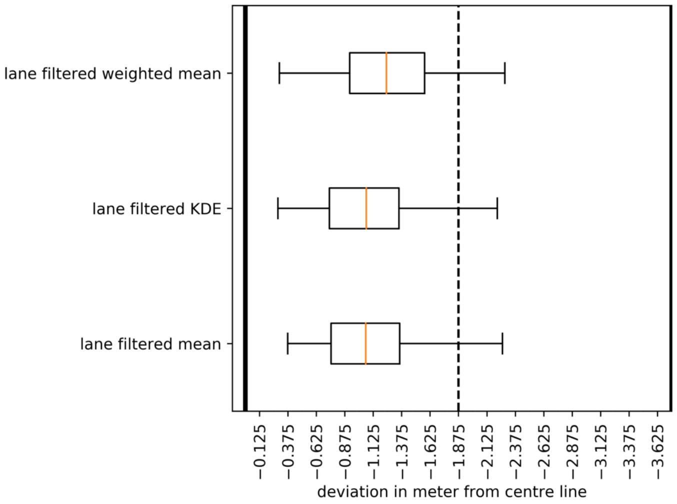


Figure 78: Comparison between weighting and non-weighting algorithms using all available pre-lane filtered traces (right lane data) [89].

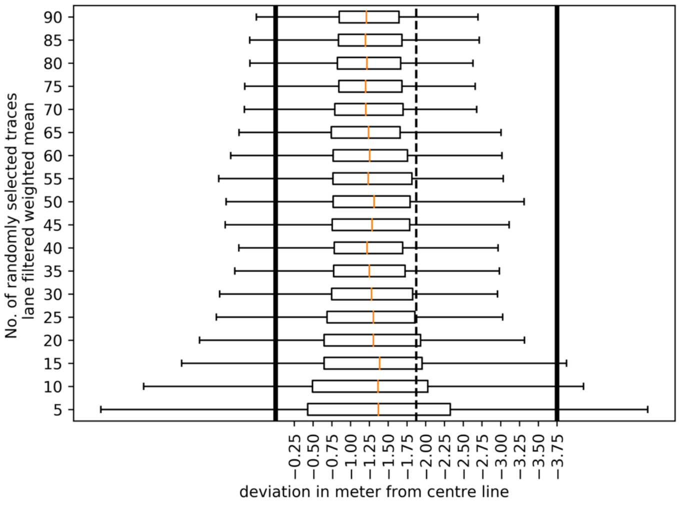


Figure 79: Influence of different amounts of available input traces on the clustering performance (right lane data) [89].

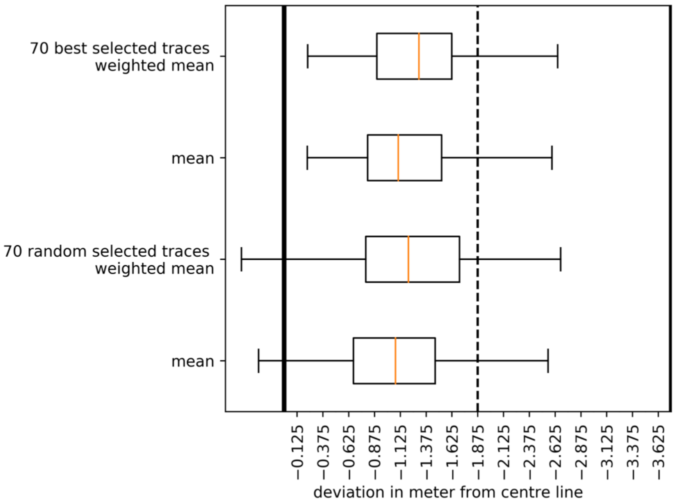


Figure 80: Performance comparison between 70 randomly selected traces and the 70 most accurate available traces (right lane data) [89].

A.8 DETAILED LISTING OF THE CELLULAR NETWORK MEASUREMENTS CONDUCTED DURING THE KO-HAF PROJECT

In Table 19 and 20 details for our dataset of the LTE[46] and UMTS[47] cellular networks, which we collected during the Ko-HAF project are summarized.

Band	Name	Earfcn DL	Downlink frequency (MHz)	Earfcn UL	Uplink frequency (MHz)	No. of measurements
1	2100	0	2110.0	18000	1920.0	126,707
1	2100	8	2110.8	18008	1920.8	502
1	2100	10	2111.0	18010	1921.0	55
1	2100	16	2111.6	18016	1921.6	77
1	2100	27	2112.7	18027	1922.7	74
1	2100	38	2113.8	18038	1923.8	145
1	2100	48	2114.8	18048	1924.8	474
3	1800+	1300	1815.0	19300	1720.0	191
3	1800+	1444	1829.4	19444	1734.4	3
3	1800+	1600	1845.0	19600	1750.0	223
3	1800+	1801	1865.1	19801	1770.1	208,249
3	1800+	1836	1868.6	19836	1773.6	53,247
7	2600	2850	2630.0	20850	2510.0	458,090
20	800 DD	6200	796.0	24200	837.0	16
20	800 DD	6300	806.0	24300	847.0	2,365,760
20	800 DD	6400	816.0	24400	857.0	172

Table 19: Overview about the different LTE frequency bands, measured during the test drives performed in the Ko-HAF project.

---

[46] http://niviuk.free.fr/lte_band.php (Last accessed on August 1, 2019)

[47] http://niviuk.free.fr/umts_band.php (Last accessed on August 1, 2019)

Band	Name	Uarfcn DL	Downlink frequency (MHz)	Uarfcn UL	Uplink frequency (MHz)	No. of measurements
1	2100	10564	2112.8	9614	1922.8	304,991
1	2100	10588	2117.6	9638	1927.6	34,487
1	2100	10612	2122.4	9662	1932.4	9056

Table 20: Overview about the different UMTS frequency bands, measured during the test drives performed in the Ko-HAF project.

A.9 DETAILED CALCULATION FOR FRANKFURT-NIEDERRAD SCENARIO

In the following the detailed calculations of the data rates required for our considered worst-case communication scenario in Frankfurt-Niederrad are conducted.

Preconditions:
Map tile size = 66 kByte
Map tile dimensions[48] =

$$1 \text{ degree latitude } * 1 \text{ degree longitude in Germany } \approx 1,190 \text{ m (latitude) } * 1,850 \text{ m (longitude)}$$

We assume all 15,158 cars during the rush hour to travel at a speed of 100 km/h and simplify the driving direction by assuming only traveling in parallel to the earth meridian. In the designated area 6 individual cells provide their network capacity in parallel to each other to the clients/cars.

Calculation:
Required download data rate:

$$66 \text{ kByte}/1.190 \text{ km} \approx 55.5 \text{ kByte/km}$$

Required upload data rate:
low sampling rate:

$$250 \text{ kByte/km}$$

high sampling rate:

$$1035 \text{ kByte/km}$$

Vehicles per cell in 3600 s =

$$15,158/6 = 7579/3$$

Time required for one car to traverse the cell area=

$$3.2 \text{ km } * 3600 \text{ s}/100 \text{ km} = 115,2 \text{ s}$$

Amount of data generated / required per vehicle during the traverse of one cell =

$$3.2 \text{ km } * \text{data rate of vehicle [kByte/km]}$$

---

[48] http://www.iaktueller.de/exx.php (Last accessed on August 1, 2019)

$$\Rightarrow$$

Required data rate per vehicle =

$$3.2\text{ km } * \text{ data rate of vehicle [kByte/km]}/115,2\text{ s}$$

Number of vehicles in parallel in one cell at any given time =

$$15,158\text{ vehicles }/6 * 115,2\text{ s}/3600\text{ s } \approx 80.84\text{ vehicles}$$

$$\Rightarrow$$

Required data per cell tower =

$$3.2\text{ km } * \text{ data rate of vehicle [kByte/km] }/115,2\text{ s } * 80.84$$

$$\Rightarrow$$

Required download data rate =

$$3.2\text{ km } * 55.5\text{ kByte/km }/115,2\text{ s } * 80.84 \approx 124.6\text{ kByte/s } \approx 0.997\text{ Mbit/s}$$

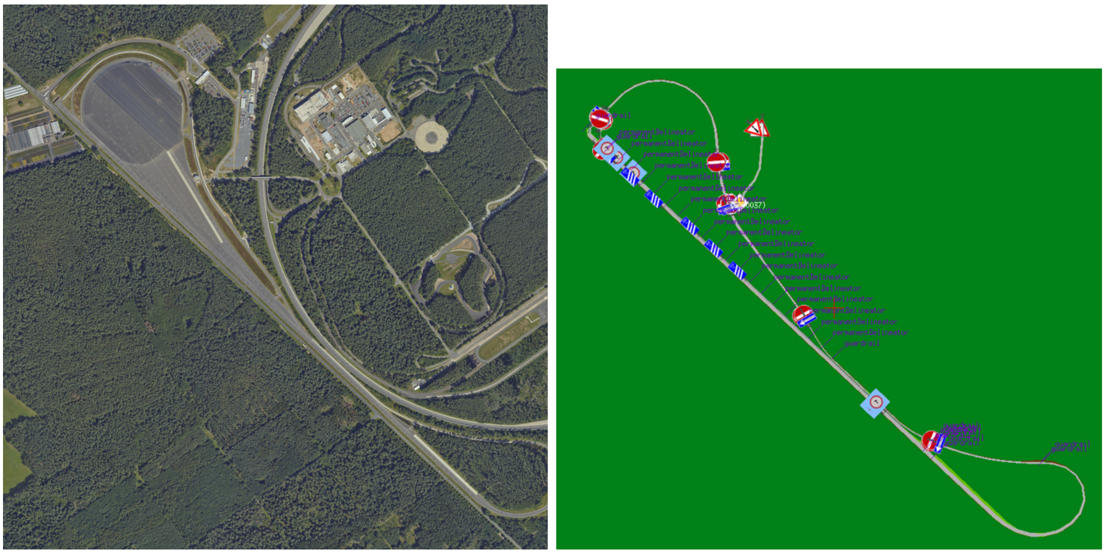
Required upload data rate:
low sampling rate:

$$3.2\text{ km } * 250\text{ kByte/km }/115,2\text{ s } * 80.84 \approx 561.4\text{ kByte/s } \approx 4.49\text{ Mbit/s}$$

high sampling rate:

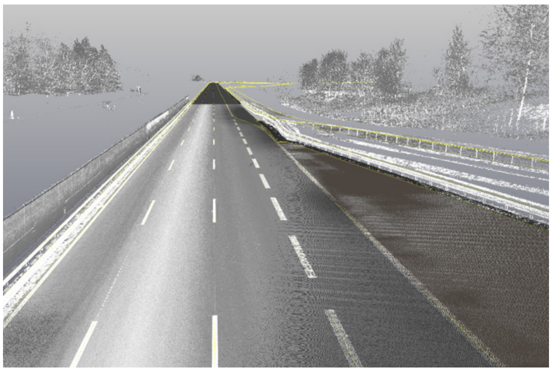
$$3.2 * 1035\text{ kByte/km }/115,2\text{ s } * 80.84 \approx 2324.15\text{ kByte/s } \approx 18.59\text{ Mbit/s}$$

A.10 EVALUATION OF OPEL PROVING GROUND IN RODGAU-DUDENHOFEN



(a) Aerial view - ©Aerowest GmbH[49]

(b) HD Map representation in Open Drive format ©Ko-HAF, 3D Mapping Solutions GmbH



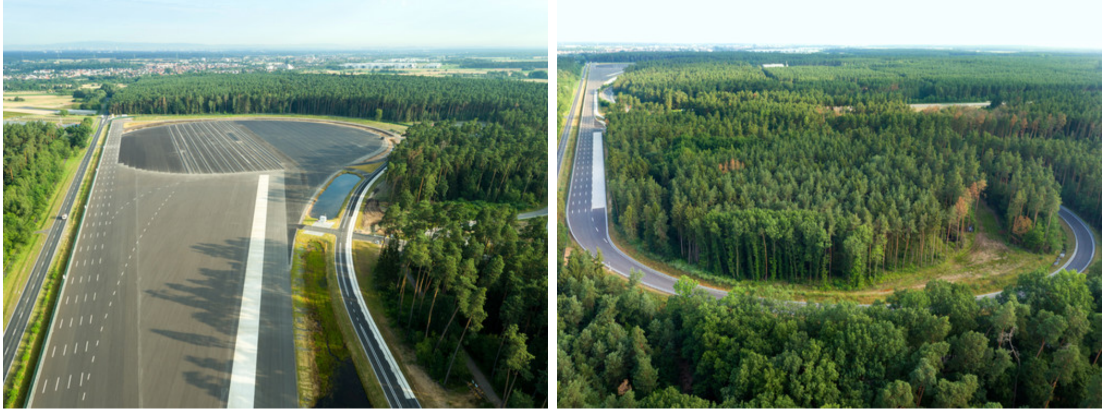
(c) Visualisation of Laser Scanner data used for HD Map generation - ©3D Mapping Solutions GmbH

Figure 81: Opel proving ground in Rodgau-Dudenhofen.

The Opel proving ground in Rodgau Dudenhofen as described in Section 6.1.1 and further illustrated by the images of Figure 81 is our second environment for which we collected cellular network quality information from a fleet of self-driving prototype vehicles. In contrast to the highways around Frankfurt the cell tower, which was dedicated to serve the Opel proving ground, as shown in Figure 83, was only equipped with 3G network technology at the start of the Ko-HAF project. An LTE signal could only be obtained occasionally when being served by one of the surrounding

---

[49] https://ko-haf.de/fileadmin/user_upload/media/abschlusspraesentation/14_Ko-HAF_Continuous-Updating-of-Backend-HD-Map-Data.pdf (Last accessed on August 1, 2019)



(a) Skidpad in the north of the track (b) Steep turn in the south of the track

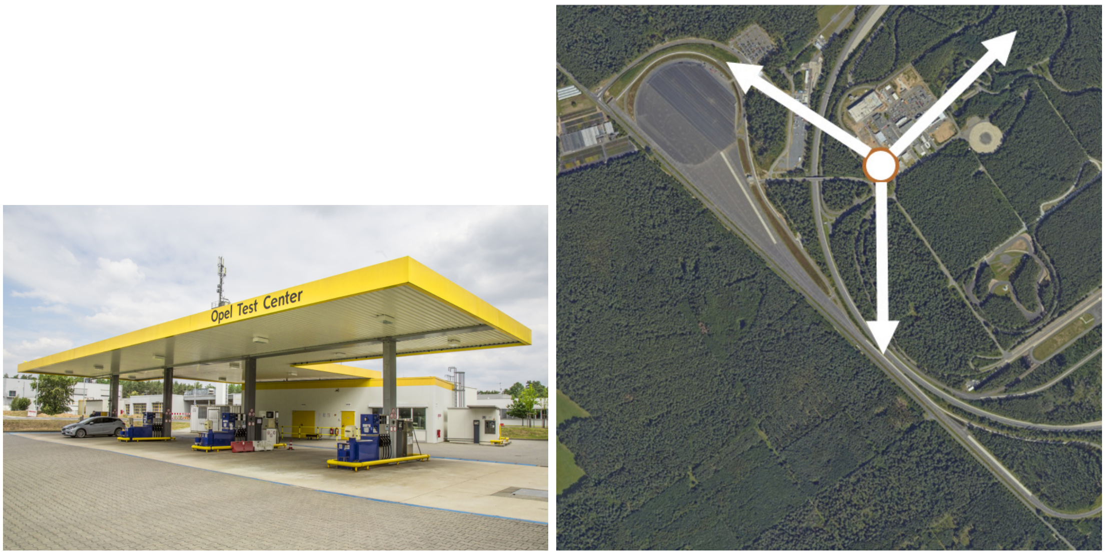
Figure 82: Aerial images of the north and the south section of the highway areal on the Opel proving ground in Rodgau-Dudenhofen. ©Opel Automobile GmbH[50]

cells. To improve the network's capacity for the successful testing of the self-driving prototype vehicles the cell tower site was upgraded to the LTE technology during the runtime of the Ko-HAF project (at the weekend between 26.01.2018 and 28.01.2018). As highly interesting evaluation result in consequence we were able to collect cellular measurements before and after this grid extension. The overall impact of the grid extension on the network's throughput performance is summarized in Figure 84. The obtained measurement values for the Round Trip Time (RTT) and the Reference Signal Received Power (RSRP) are presented by the Figures 85 and 86 respectively. As expected all aggregation results show a significant positive impact on the achieved performance of all network quality indicators. However there are still variations in terms of the performance. This is furthermore note worthy considering the rather small area of the test site in comparison to the area around Frankfurt investigated previously.

The achieved throughput values and the received RSRP value for example are better in the area around the skid pad in the north (Fig. 82a) and rather poor in the area of the steep curve in the south of the track (Fig. 82b). Possible reasons therefore likely include environmental conditions as described in Section 1.2. The skid pad represents a large flat tar area, without any obstructing buildings or similar objects, which provides nearly perfect reception conditions for the cellular signal. The steep curve in the south of the track in contrast is surrounded by high trees, which might significantly damp the received cellular signal especially when the leafs of the trees are wet. A further indicator supporting this assumption is the experienced rather poor GNSS localization capabilities of the self-driving vehicles in this area as reported by the Ko-HAF project partners during the test drives. As for example the GPS system is using similar frequencies (L1-band ~ 1.57 GHz, L2-band ~ 1.23 GHz) [103] it likely experiences similar damping factors.

---

[50] https://www.auto-medienportal.net/artikel/detail/37152 (Last accessed on August 1, 2019)



(a) Central location of the serving LTE cell tower on top of a chimney near the gas station on the proving ground. ©Opel Automobile GmbH[51]

(b) Aerial view of the cell tower's location with the indicated main transmission directions of its antennas. ©Aerowest GmbH

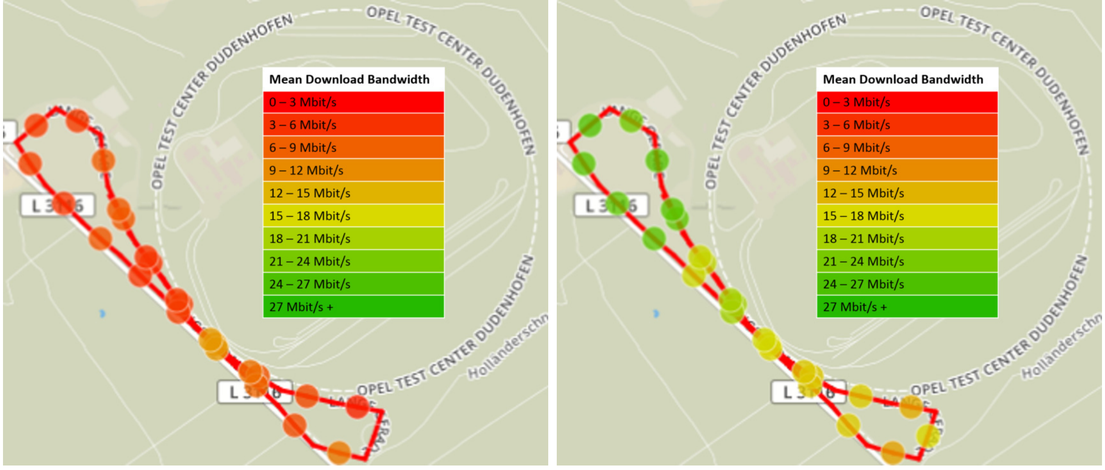
Figure 83: Location of the LTE cell tower on the Opel proving ground in Rodgau-Dudenhofen.

In parallel to our personal measurement campaign Vodafone also conducted their own reference measurements of the Reference Signal Received Power at the Opel proving ground after the executed grid extension. To our satisfaction the obtained measuring results of these test drives as summarized by Figure 86c are closely resembled by our own measurements (compare with Figure 86b). Additional measurements of the complete proving ground are shown in Figures 87 and 88.

We consider this as a strong emphasis in support of the related and our personal scientific work that vehicular probes can be used to collect reliably cellular network quality information to be further used for the optimization of the overall achieved network quality experience as described in Section 6.2.

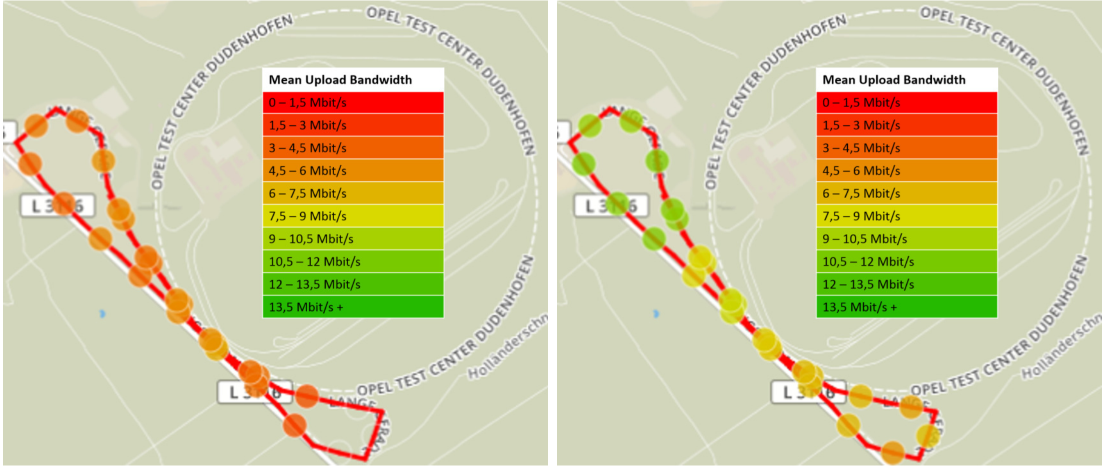
---

[51] https://awrmagazin.de/wp-content/uploads/2016/09/foto-opel-dudenhofen-6.jpg (Last accessed on August 1, 2019)



(a) Mean download bandwidth before grid extension.

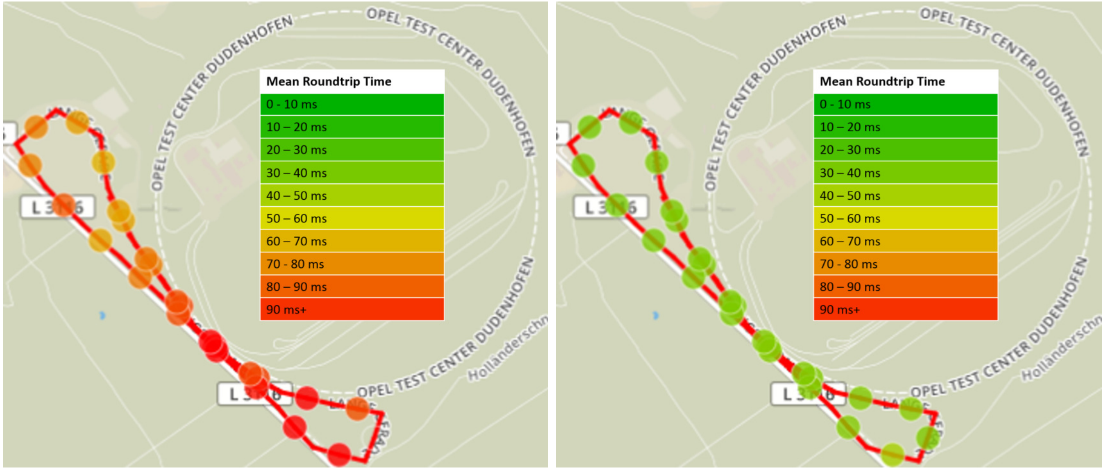
(b) Mean download bandwidth after grid extension.



(c) Mean upload bandwidth before grid extension.

(d) Mean upload bandwidth after grid extension.

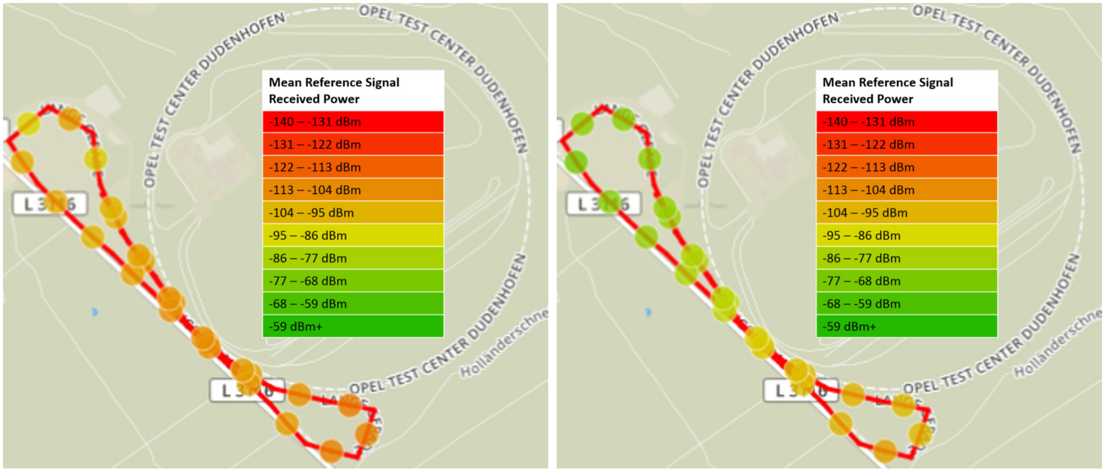
Figure 84: Comparison of mean average download and upload bandwidth of the provider Vodafone as measured at the Opel proving ground in Rodgau Dudenhofen before (07.11.2017-29.01.2018) and after (29.01.2018-20.09.2018) the LTE grid extension. Map data ©OpenStreetMap contributors



(a) Mean round trip time before grid extension.

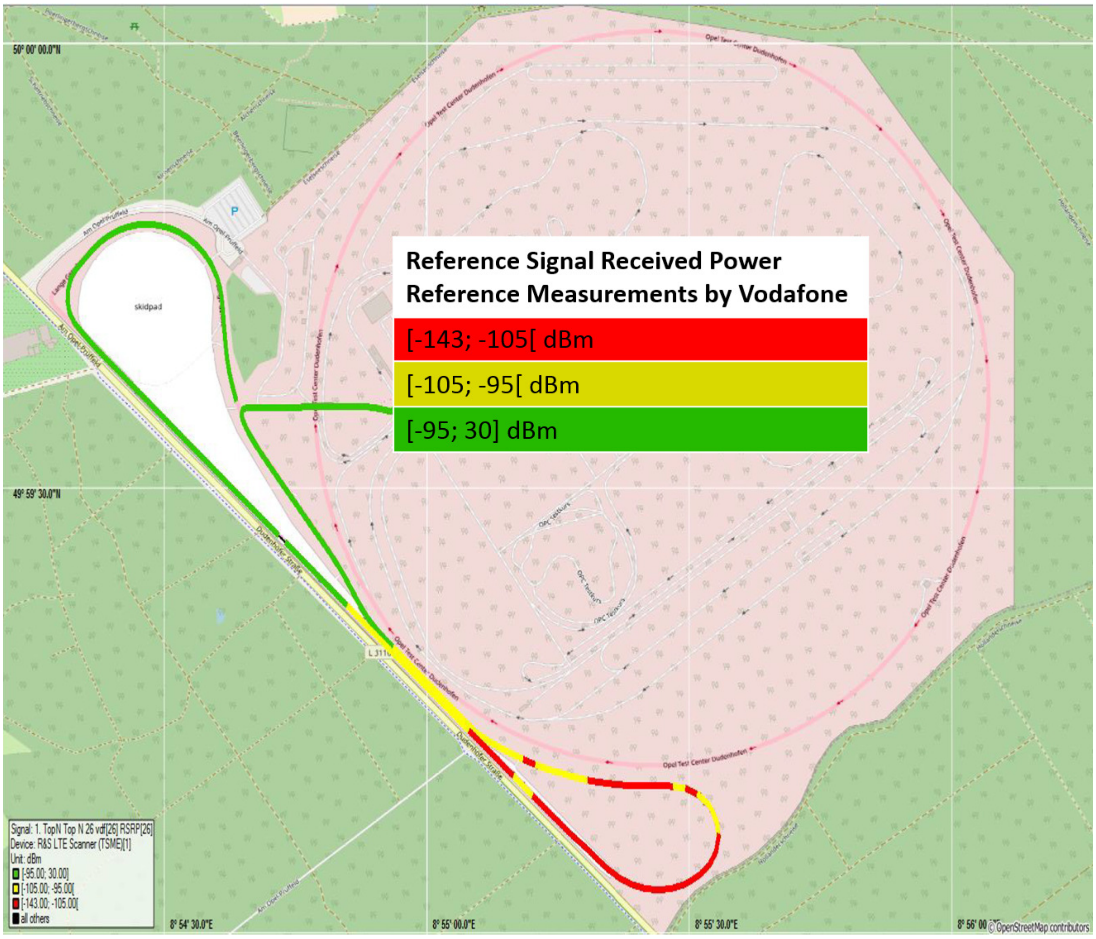
(b) Mean round trip time after grid extension.

Figure 85: Comparison of the achieved Round Trip Time values before (07.11.2017-29.01.2018) and after (29.01.2018-20.09.2018) the LTE grid extension. Map data ©OpenStreetMap contributors



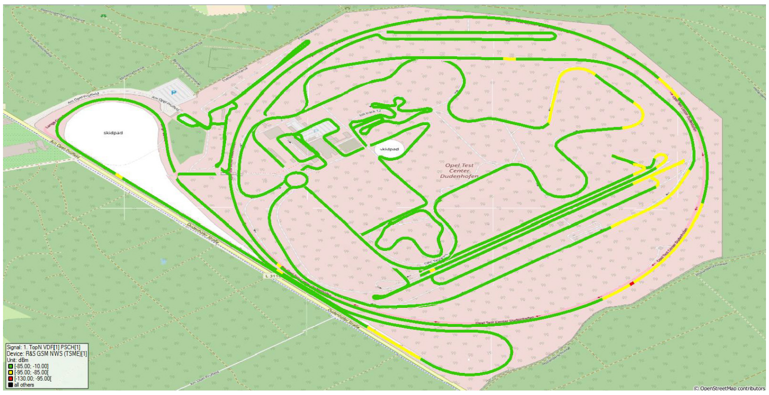
(a) Measured Reference Signal Received Power at the first day of the measuring campaign on the site (07.11.17).

(b) Measured Reference Signal Received Power at the last day of the measuring campaign on the site (20.09.18).

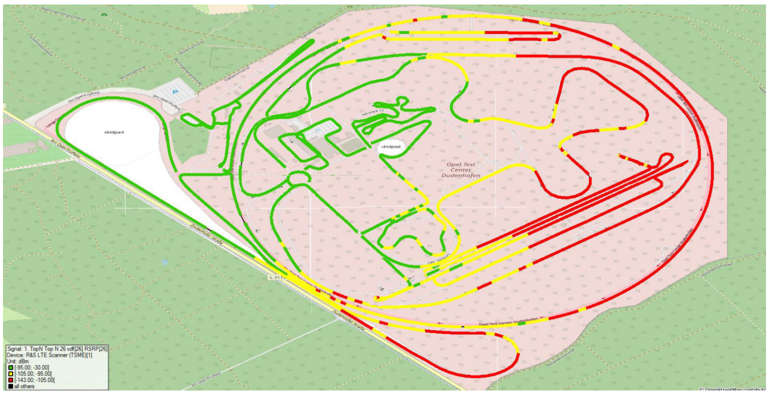


(c) Reference measurements of the RSRP value of the LTE network conducted for Vodafone after the grid extension (12.04.2018).

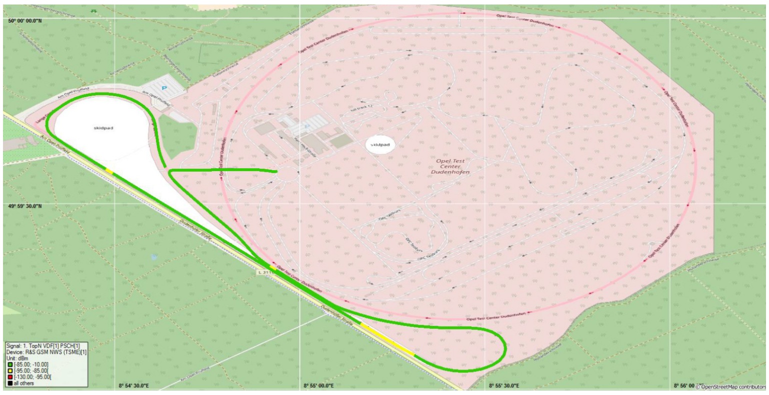
Figure 86: Comparison of the Reference Signal Received Power measured at the first and the last day of the cellular data collection campaign in comparison to the reference measurements conducted by Vodafone after the grid extension. ©Vodafone GmbH and FMB Engineering GmbH, map data ©OpenStreetMap contributors


Signal: 1. TopN VDF[1] PSCH[1]
Device: R&S GSM NWS (TSME)[1]
Unit: dBm
🟩 [-85.00; -10.00]
🟨 [-95.00; -85.00[
🟥 [-130.00; -95.00[
⬛ all others
Opel Test Center Dudenhofen
skidpad
© OpenStreetMap contributors

(a) Complete Opel proving ground - GSM (PSCH)

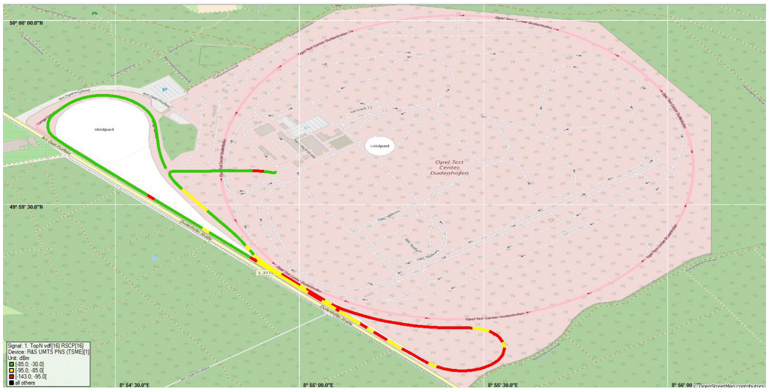

Signal: 1. TopN Top N 26 vdf[26] RSRP[26]
Device: R&S LTE Scanner (TSME)[1]
Unit: dBm
🟩 [-95.00; -30.00]
🟨 [-105.00; -95.00[
🟥 [-143.00; -105.00[
⬛ all others
Opel Test Center Dudenhofen
skidpad
© OpenStreetMap contributors

(b) Complete Opel proving ground - LTE 800 (RSRP)

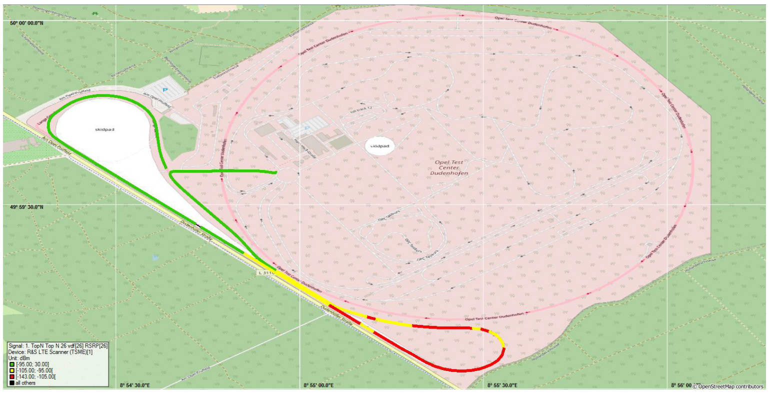


(c) Highway section on the Opel proving ground - GSM (PSCH)

Figure 87: Reference measurements of the experienced signal strength of the various available network technologies (GSM, UMTS, LTE) on the complete Opel proving ground in Rodgau-Dudenhofen. The measurements were conducted by Vodafone after the LTE network expansion (12.04.2018). ©Vodafone GmbH and FMB Engineering GmbH



(a) Highway section on the Opel proving ground - UMTS (RSCP)



(b) Highway section on the Opel proving ground - LTE 800 (RSRP)

Figure 88: Continued reference measurements of the experienced signal strength of the various available network technologies (GSM, UMTS, LTE) on the complete Opel proving ground in Rodgau-Dudenhofen. ©Vodafone GmbH and FMB Engineering GmbH

A.11 SIMULATED SUMO SCENARIOS

The following figures illustrate the extend of several scenarios, which have been simulated in this thesis. Namely they are Berlin (Fig. 89), Cologne (Fig. 90) and Luxembourg (Fig. 91).

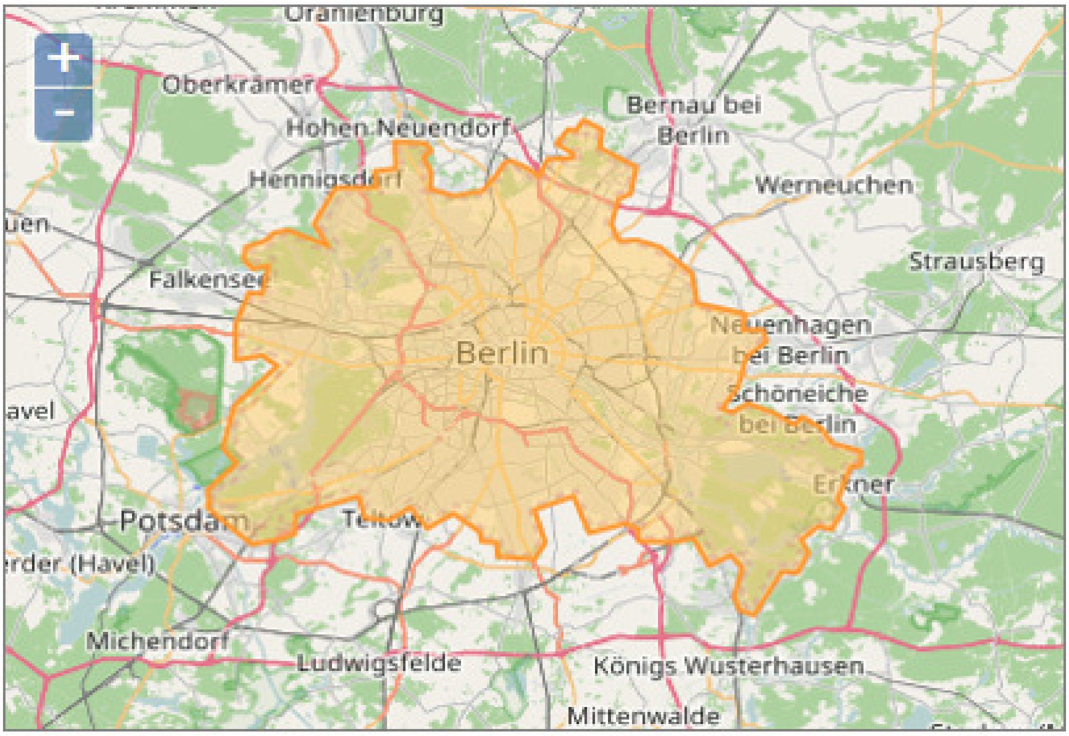
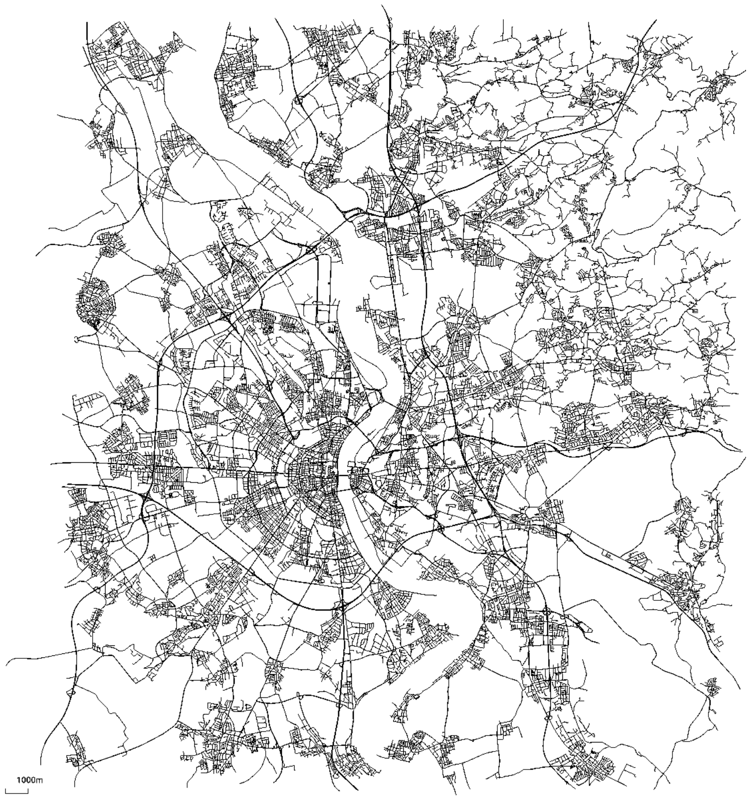


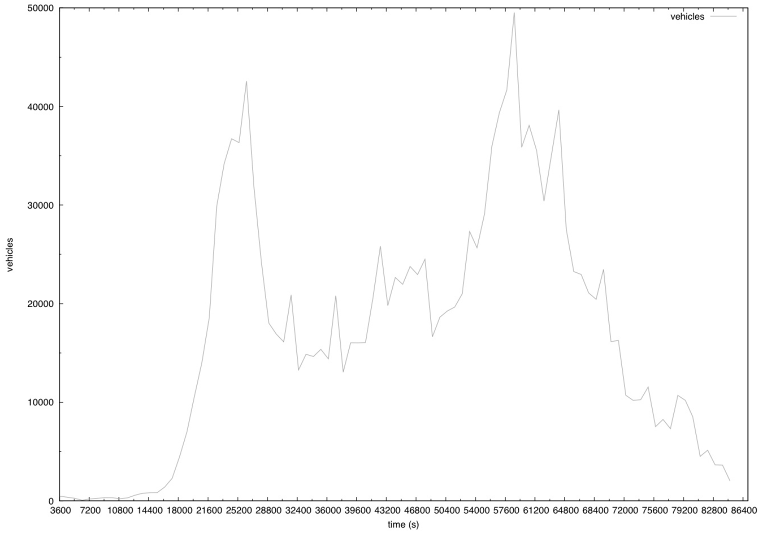
Figure 89: Visualization of the extend of the OpenStreetMap data of Berlin used for the evaluation of the Dynamic Map Update protocol. ©OpenStreetMap contributors[52]

---

[52] http://download.geofabrik.de/europe/germany/berlin.html (Last accessed on August 1, 2019)



(a) Overview about the scenario of the TAPAS Cologne dataset .



(b) Overview about the traffic pattern distribution over the full 24 hours of the TAPAS Cologne dataset.

Figure 90: Visualization of the TAPAS Cologne dataset for the traffic simulator SUMO. The dataset covers a full day of traffic in and around the city of Cologne Germany. For our simulation in Section 6.3.4 we relied on the traffic pattern between 6 and 9 o'clock in the morning. (Source: https://sumo.dlr.de/wiki/Data/Scenarios/TAPASCologne (Last accessed on August 1, 2019))

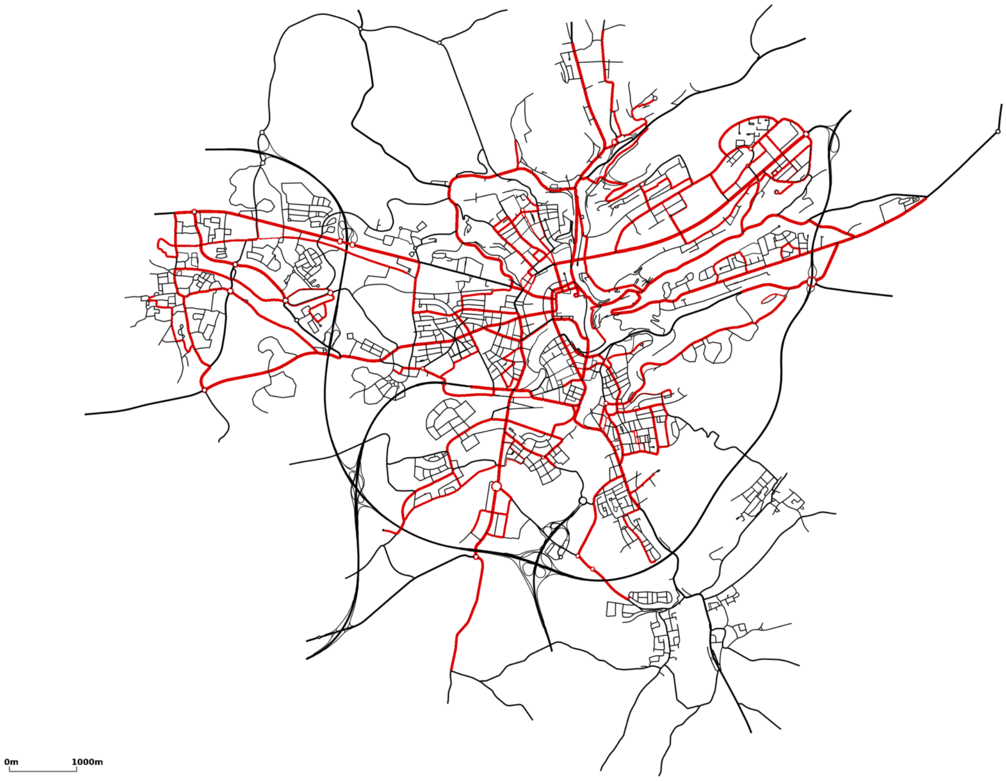


Figure 91: Visualisation of the LUST dataset [109] of the area of Luxembourg for the traffic simulator SUMO. The streets where buses are roaming in the map are indicated by red color. (Source: https://github.com/lcodeca/LuSTScenario/blob/master/docs/img/LuST_busCoverage.png (Last accessed on August 1, 2019))

A.12 DETAILED LIST OF CELLULAR NETWORK QUALITY PARAMETERS COLLECTED THROUGH THE CUSTOM ANDROID MEASURING APPLICATION

Our custom Android application relies on two different interfaces to retrieve cellular network quality parameters: i.) the open Android APIs and ii.) the quality information directly retrieved from the smartphones LTE chipset through the MobileInsight application. The following Tables 21 and 22 for Android and 23 and 24 for MobileInsight summarize all these parameters.

Provided by Android APIs

measurementtime	Time when the measurement was executed
ci	Cell ID of the currently serving cell tower (eNodeB)
pci	Physical cell ID of the serving antenna (cell) of the current eNodeB (cell tower) - commonly an eNodeB is equipped with three distinct antennas to achieve a 360° coverage of the surrounding area.
tac	Tracking Area code - identifying an area of several eNodeBs
networktype	Current used cellular technology, e.g. LTE, UMTS, HSDPA, EDGE, ...
mcc	Mobile Country Code - identifying the country of where the operating cell tower is located (262 - Germany)
mnc	Mobile Network Code - identifying the cellular network operator of the currently serving cell tower
earfcn	Evolved-UTRA Absolute Radio Frequency No. - specifying a particular carrier transmission frequency - without clarification of the used bandwidth

Table 21: List of parameters provided to the Connectivity Map Client by the Android APIs.

asulevel	ASU level of the currently serving cell tower
timing advance	timing advance value for LTE - value to ensure timely syncronisation of the transmission process between cell tower and end device
provider	Human readable name of the currently serving provider
signalstrength	Signal strength of the current connection in dBm
rsrp	Reference Signal Received Power in dBm
rsrq	Reference Signal Received Quality
rssnr	Reference Signal Signal-To-Noise Ratio
cqi	Channel Quality Indicator
speed	Current traveling speed of the vehicle
longitude/latitude	Location of the smartphone in WGS 84 coordinates (longitude and latitude)
tcp rtt	Round Trip Time estimation of the TCP connection of the control channel as a measure related to the latency of the connection
throughput	Measured throughput of the current connection
device name	Device name - used for device specific filtering
imei	International Mobile Equipment Identity - unique identification number if several devices of the same type are used throughout the measuring process
packetloss	Amount of lost packets during the sending process - retrieved from packet sequence number identification

Table 22: Continued list of parameters provided to the Connectivity Map Client by the Android APIs.

Provided via the MobileInsight application

qualitytimestamp	Time when the value was provided by the chipset
cellid	Cell ID of the currently serving cell tower (eNodeB)
mcsindex	Modulation and Coding Scheme Index - identifying the currently used Modulation and Coding Scheme of the connection (e.g. QPSK, 16 QAM, 64 QAM)
tbs	Transport Block Size
rbs	Resource Block Size
throughput	Calculated throughput estimate based on the time between two received values and the received Transport Block Size
rsrp	Reference Signal Received Power in dBm
rsrq	Reference Signal Received Quality
rssi	Received Signal Strength Indication (RSSI) in dBm
cqi	Channel Quality Indicator
earfcn	Evolved-UTRA Absolute Radio Frequency No. - specifying a particular carrier transmission frequency - without clarification of the used bandwidth
speed	Current traveling speed of the vehicle
longitude/latitude	Location of the smartphone in WGS 84 coordinates (longitude and latitude)

Table 23: Complete list of parameters provided to the Connectivity Map Client by the MobileInsight application for both the uplink and downlink similarly.

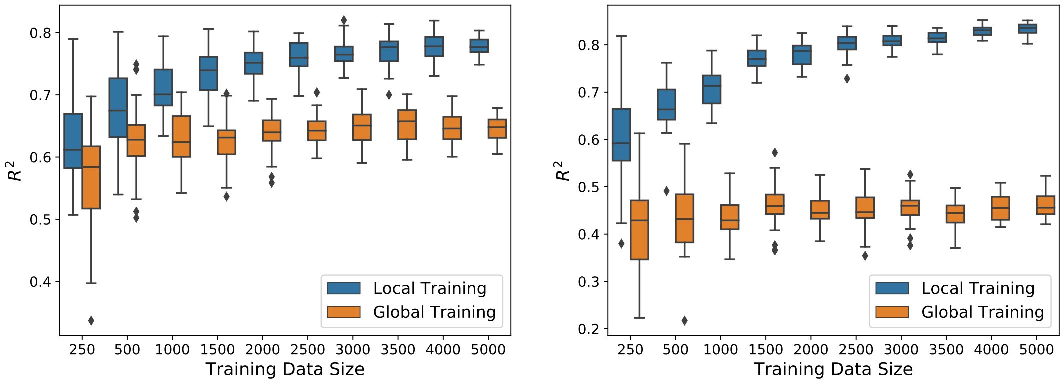
Additionally to the same values as provided in the upload direction further provided parameters in the download direction are as follows.

mimo	MIMO index - indicating wether or not Multiple Input Multiple Output data transmissions using several antennas was available at the time the value was retrieved
rnti	C-RNTI - a unique identifier used for identifying RRC Connection and scheduling which is dedicated to a particular user end device (UE)
caindex	Carrier Aggregation Index [0,1] - indicating wether or not Carrier Aggregation was available at the time the value was retrieved

Table 24: List of additional parameters provided to the Connectivity Map Client by the MobileInsight application only for the downlink.

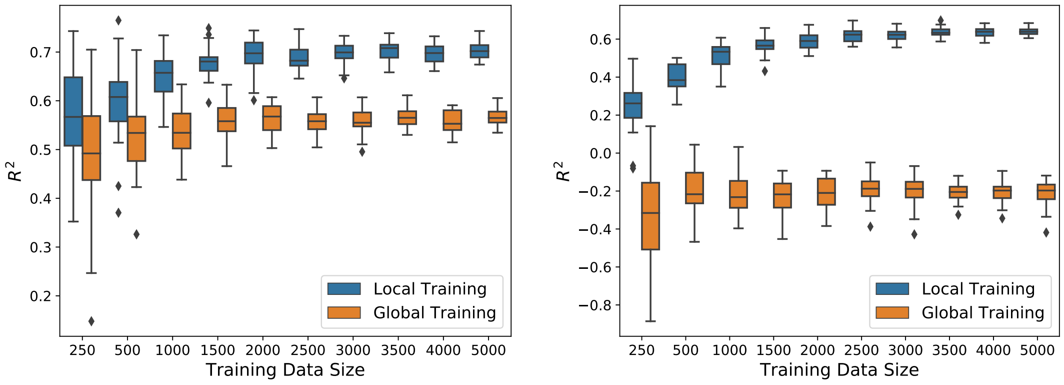
A.13 LOCALIZED VS GLOBALIZED TRAINING BASED ON MOBILEINSIGHT DATA FROM PROVIDER B AND C

The following Figures 92, 93 and 94 show our evaluation results for the throughput prediction using either globalized or localized training data of the Providers B and C.



(a) Downlink - Cell A

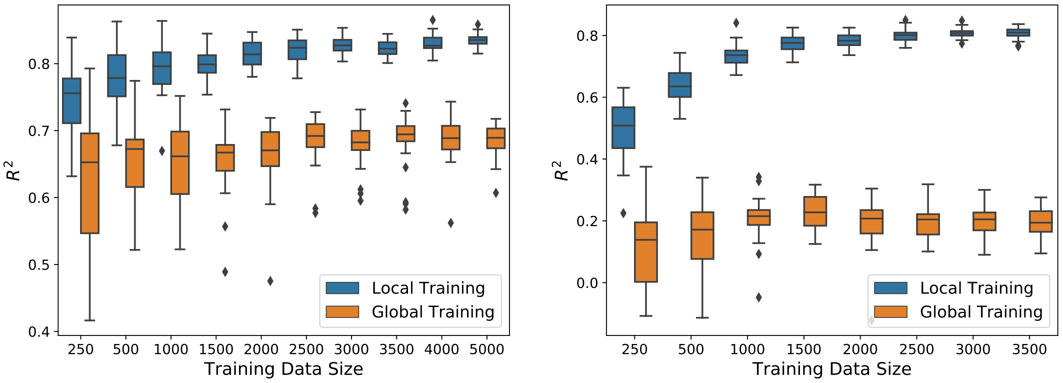
(b) Uplink - Cell A



(c) Downlink - Cell B

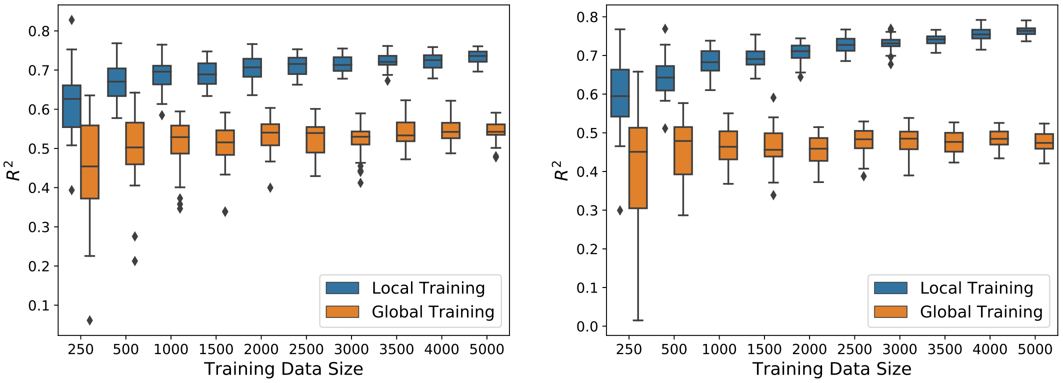
(d) Uplink - Cell B

Figure 92: Performance comparison between localized and global training data of the first and second most often measured cells of provider B.



(a) Downlink - Cell C

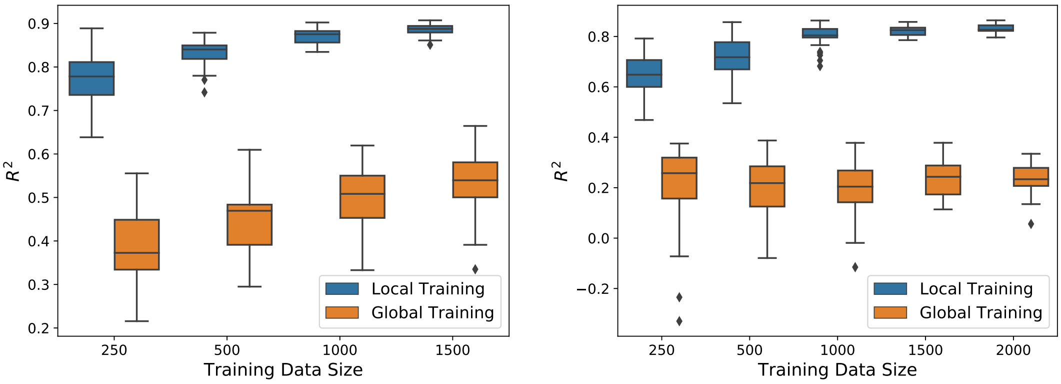
(b) Uplink - Cell C



(c) Downlink - Cell D

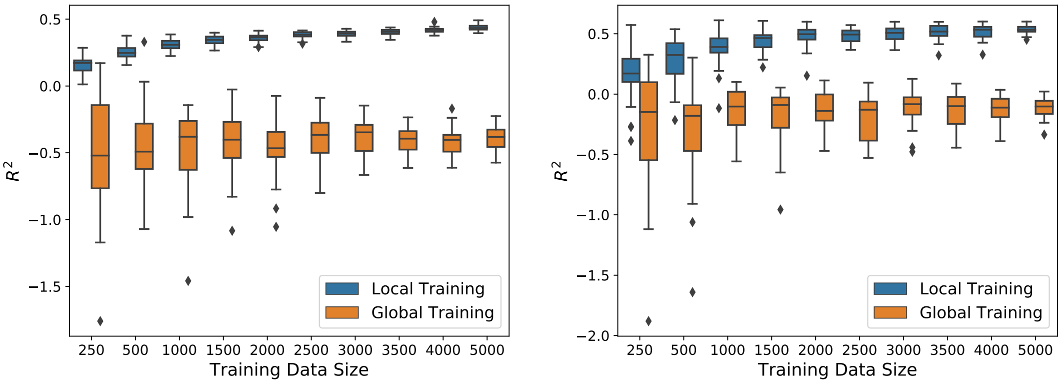
(d) Uplink - Cell D

Figure 93: Continued performance comparison between localized and global training data of the third and fourth most often measured cells of provider B.



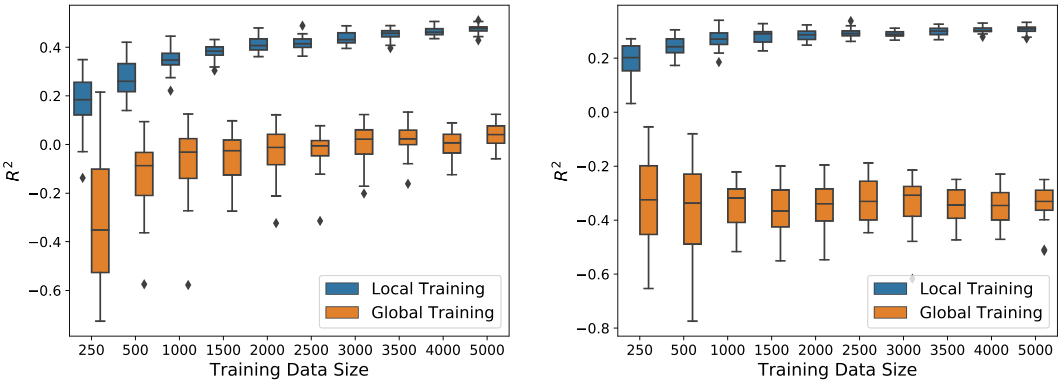
(a) Downlink - Cell A

(b) Uplink - Cell A



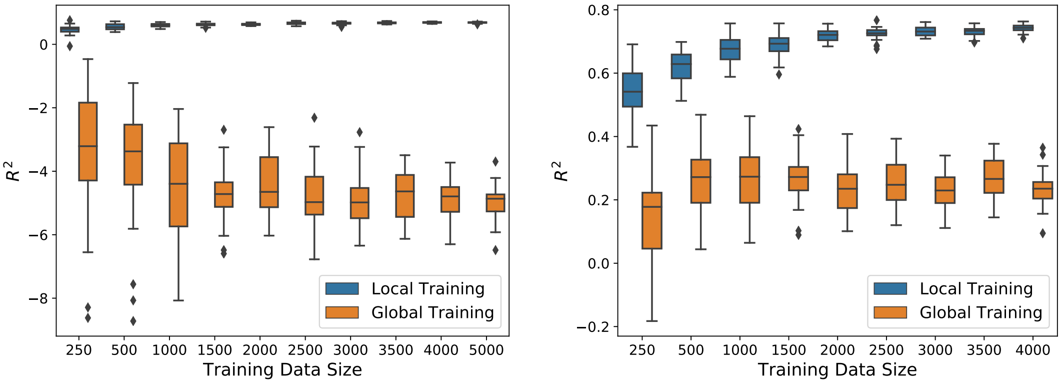
(c) Downlink - Cell B

(d) Uplink - Cell B



(e) Downlink - Cell C

(f) Uplink - Cell C



(g) Downlink - Cell D

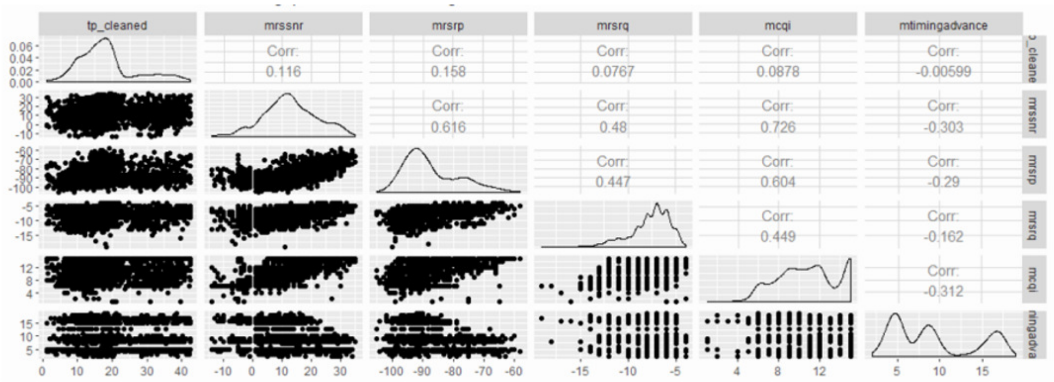
(h) Uplink - Cell D

Figure 94: Performance comparison between localized and global training data of the four most often measured cells of provider C.

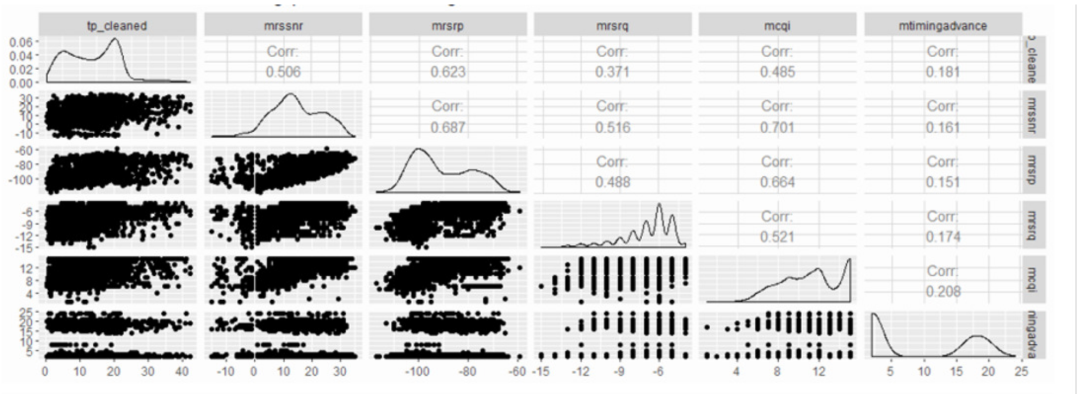
A.14 DETAILED CORRELATION PLOTS FOR THE ONLINE ESTIMATION

In the following the detailed correlation matrices (Pearson correlation coefficients) between the measured throughput values and the network performance parameters, which are used as prediction features for our online estimator are summarized. Shown are the obtained values for the four most often measured cell IDs (A-D).

First measuring campaign



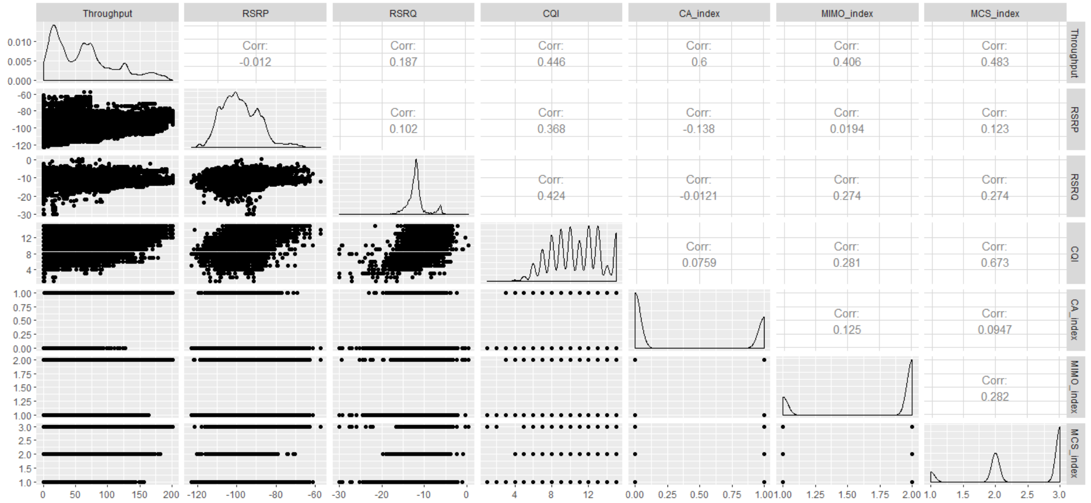
(a) Correlation coefficients for the download direction - data set of cell A



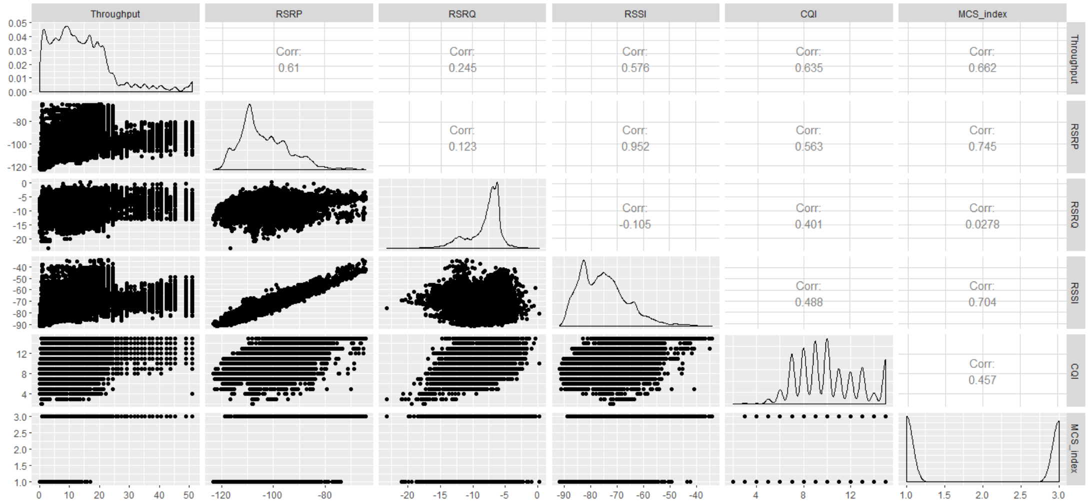
(b) Correlation coefficients for the download direction - data set of cell D

Figure 95: Achieved correlation coefficients for the cells A and D in the download direction in the first measuring campaign.

Second measuring campaign

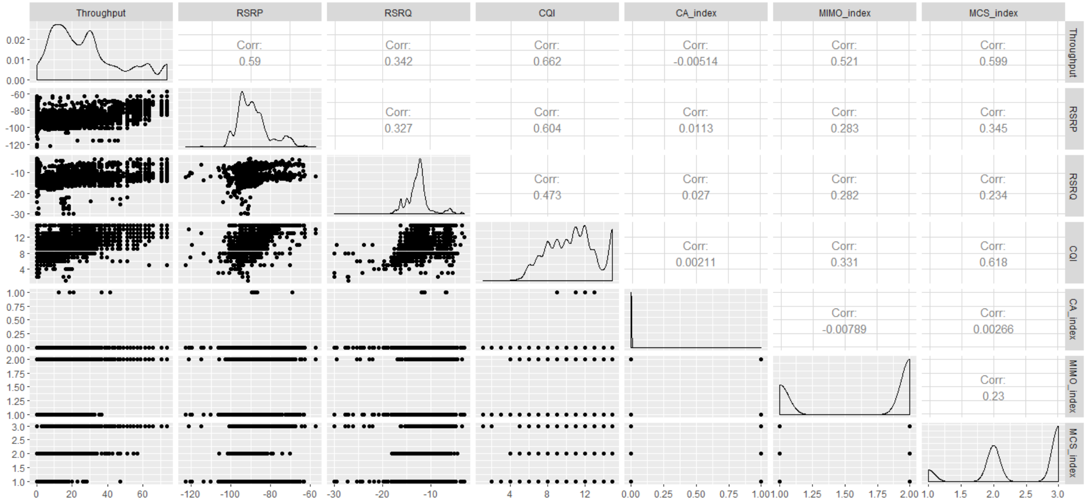


(a) Correlation coefficients for the download direction - data set of all cells together

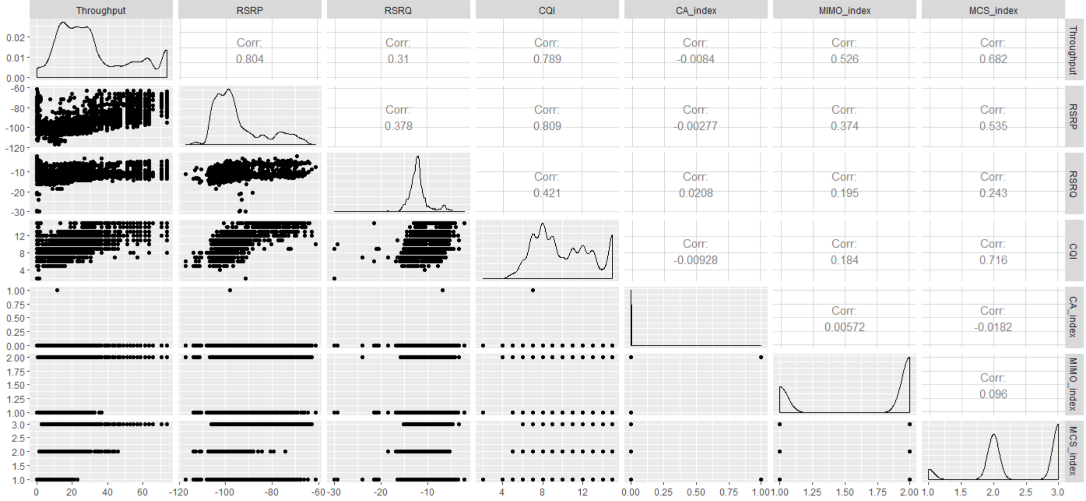


(b) Correlation coefficients for the upload direction - data set of all cells together

Figure 96: Achieved correlation coefficients for the download and the uplod direction in the second measuring campaign, when considering the complete measurement data as one training set.

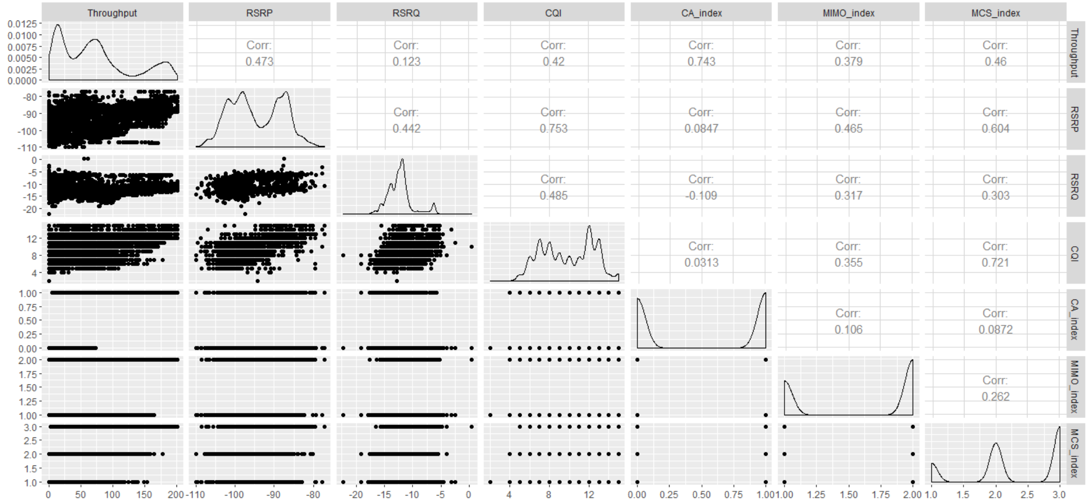


(a) Correlation coefficients for the download direction - data set of cell A

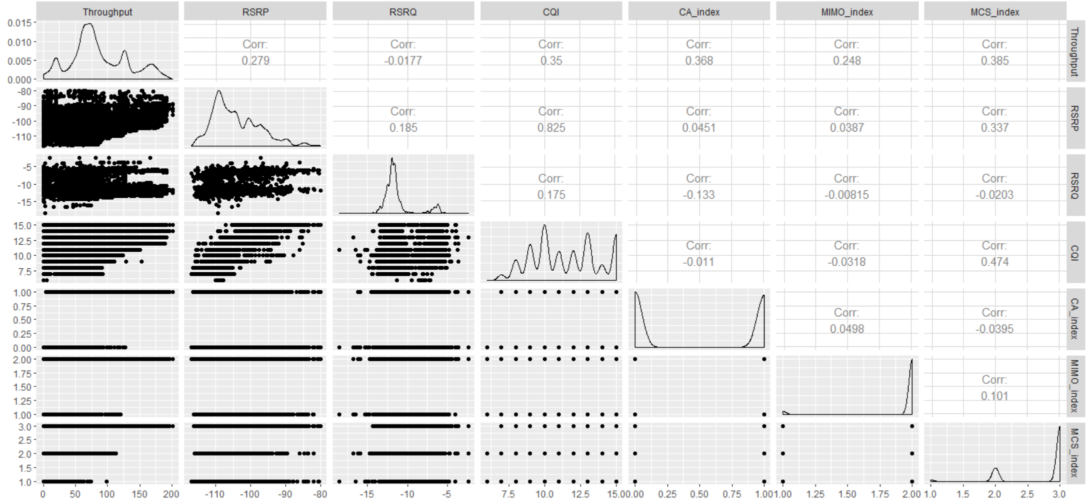


(b) Correlation coefficients for the download direction - data set of cell B

Figure 97: Achieved correlation coefficients for the download direction in the second measuring campaign, when considering only cell specific training sets.

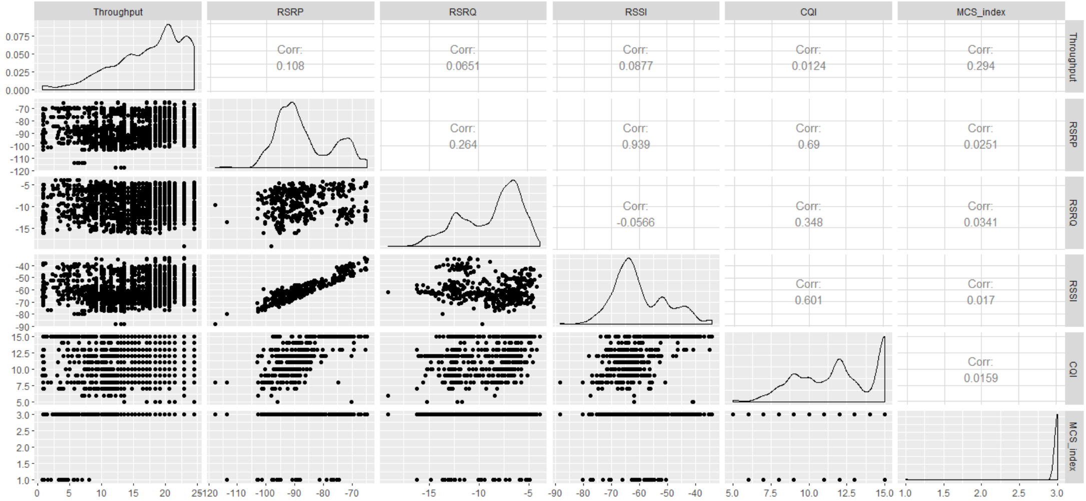


(a) Correlation coefficients for the download direction - data set of cell C

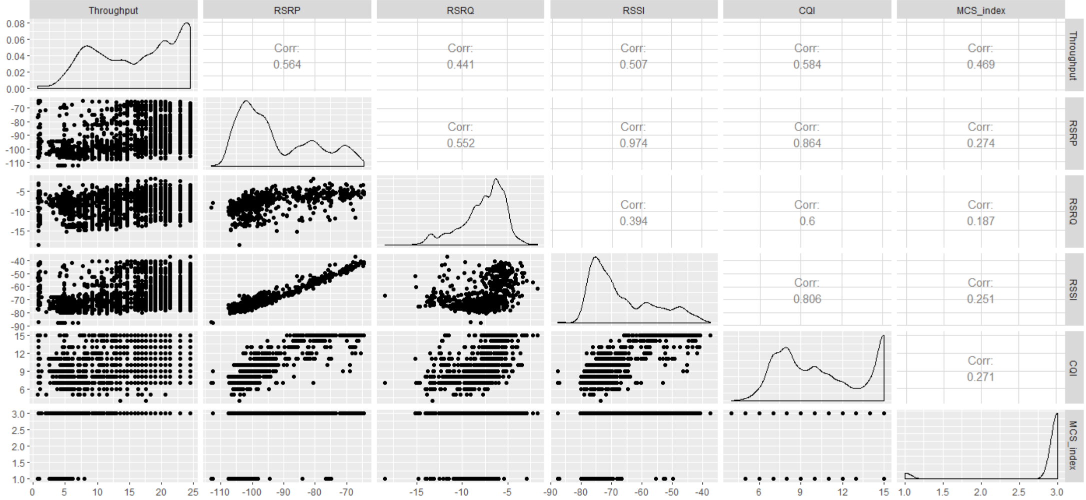


(b) Correlation coefficients for the download direction - data set of cell D

Figure 98: Continued achieved correlation coefficients for the download direction in the second measuring campaign, when considering only cell specific training sets.

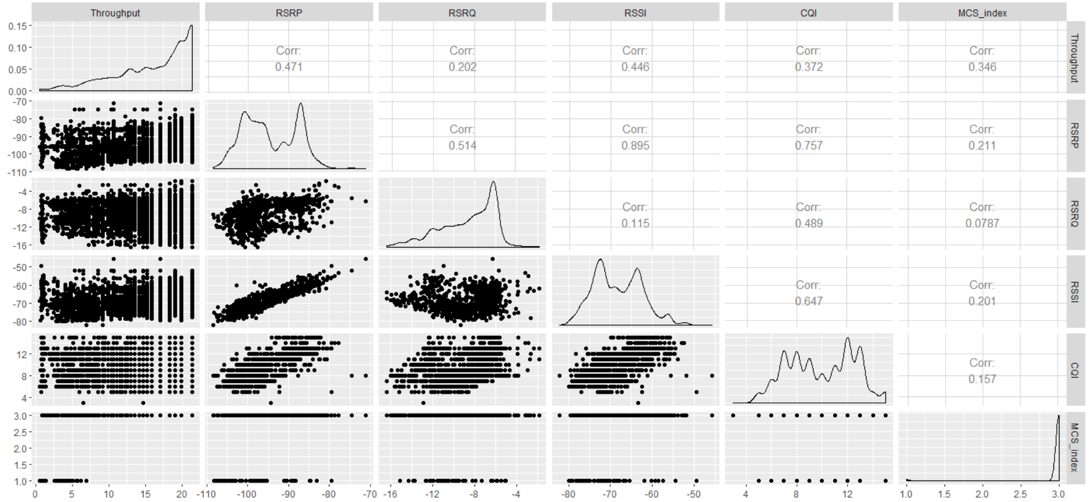


(a) Correlation coefficients for the upload direction - data set of cell A

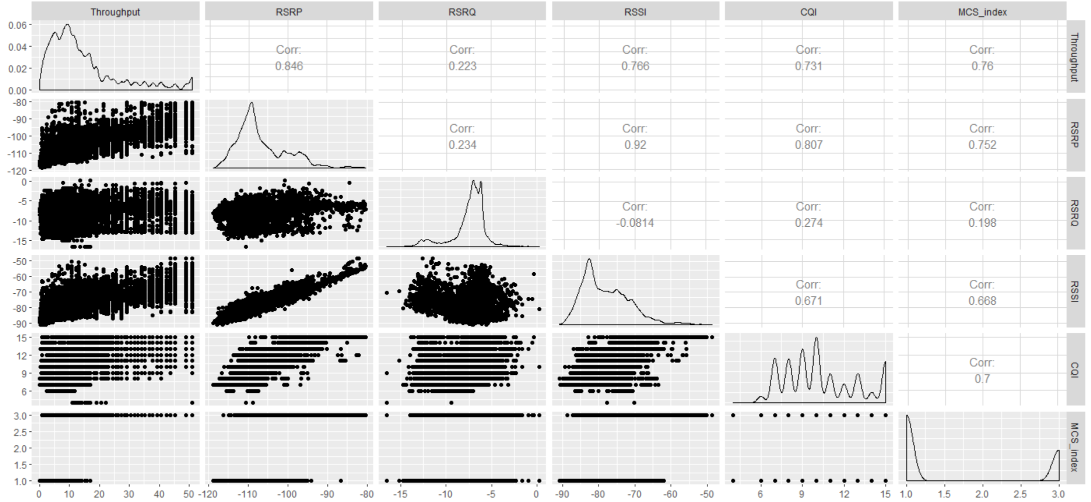


(b) Correlation coefficients for the upload direction - data set of cell B

Figure 99: Achieved correlation coefficients for the upload direction in the second measuring campaign, when considering only cell specific training sets.



(a) Correlation coefficients for the upload direction - data set of cell C



(b) Correlation coefficients for the upload direction - data set of cell D

Figure 100: Continued achieved correlation coefficients for the upload direction in the second measuring campaign, when considering only cell specific training sets.

A.15 MOBILE TEST OF IMDEA OWL CONNECTED TO CONNECTIVITYMAP CLIENT IN VEHICLE

To test and evaluate the decoding performance of Imdea OWL in a mobile scenario, we used our Connectivity Map Client application to execute throughput measurements, while the measuring equipment is moving inside of a vehicle. To validate the performance of OWL the Connectivity Map Client transmits fixed amounts of data (500 packets, each with a size of 1500 byte) over the network. Before starting the transmission process the application retrieves the currently used transmission frequency of the cell tower and uses this information to start the Imdea OWL decoding process. During the execution of the throughput measurement Imdea OWL continuously retrieves the number of currently active clients (identified via their assigned C-RNTI number) and their present network activities through the scheduled amount of resource blocks (Sec. A.3.9) per client. We selected a rather small amount of 500 data packets, with a total size of about 732 kByte, to ensure that the data transmission was completed using only one unique C-RNTI. In consequence, in a correct and unobstructed decoding procedure our test measurements should be mapable upon one specific C-RNTI and related to the correct transport block size (Sec. A.3.9) required for the transmission of our specified amount of data payload. For the verification of these two numbers we relied upon the MobileInsight interface used in our previous work (Sec. 6.4). Via the MobileInsight application we were able to retrieve the current C-RNTI, as well as the allocated transport block size of the smartphone under test. To identify issues in the decoding process related to poor signal reception, the Connectivity Map Client furthermore provides current signal strength readings (e.g. the Reference Signal Received Power) and GNSS location information to be stored with the correlating OWL LTE control channel decodings. With this configuration setup inside a vehicle, as shown in Figure 103, we performed an initial performance evaluation. We started our tests close to a cell tower as indicated by the black and white double circle in Figure 101 and 102. In total we conducted 28 measurements driving in the area around the cell tower at different distances. Accordingly we experienced different Reference Signal Received Power values during the decoding procedure. The evaluation results are summarized in Table 25. The measurements were conducted at the LTE transmission frequency of 2100 MHz.

Out of the 28 total measurements 11 measurements were identified to have matches between the decoding results of ImdeaOWL and the information retrieved by MobileInsight from the LTE chipset of the smartphone. Figure 101 visualizes the positions of successful (green) and unsuccessful (red) measurements around the cell tower. In total we could identify 61 scheduled [system frame, subframe] combinations having the same C-RNTI and allocated transport block size. The allocation of resources was also indicated by OWL throughout the measurements graphically (see e.g. Fig. 103) and correlated with our conducted throughput tests. The achieved results showcase that through our made adaptations ImdeaOWL can be quickly configured to be applied in a mobile, vehicular environment.

Total conducted measurements	28
Identified measurements with matches of C-RNTI, system frame number, sub frame number and transport block size in them	11
Total identified complete matches between ImdeaOWL and MobileInsight (same C-RNTI, system frame number, sub frame number, transport block size)	61
Reference Signal Received Power range in which the measurements where conducted	-108 to -61 dBm

Table 25: Evaluation results for the decoding comparison of ImdeaOWL and MobileInsight.

In our opinion the possible causes for the remaining 17 measurements having no direct correlation between the decoding of ImdeaOWL and MobileInsight are not related to the present signal strength conditions during the 28 measurements. The measurements cover a range of Reference Signal Received Power values between -108 dBm and -61 dBm, visualized in Figure 102. Compared to our received RSRP values in Section 6.1.5 these values indicate a medium to good signal quality. However the measurement executed under the poorest signal conditions (-108 dBm) indicated a successful match between the decodings of ImdeaOWL and the information obtained by MobileInsight. For the measurement executed at the best signal conditions (-61 dBm) we could not identify such a match. Further measurements to cover a wider signal strength range is suggested to indicate an upper limit for a successful decoding with ImdeaOWL. By investigating the detailed decoding results of ImdeaOWL and MobileInsight we identified that both sources were missing out several system frame numbers in the row of consecutive decoded frames. These are possible missing candidates for further matches. Possible reasons therefore could lie in the decoding speed of MobileInsight being limited on the tested smartphone and thus missing out important information. In consequence we advise further investigative future work to identify possible optimization potential of both software tool kits.

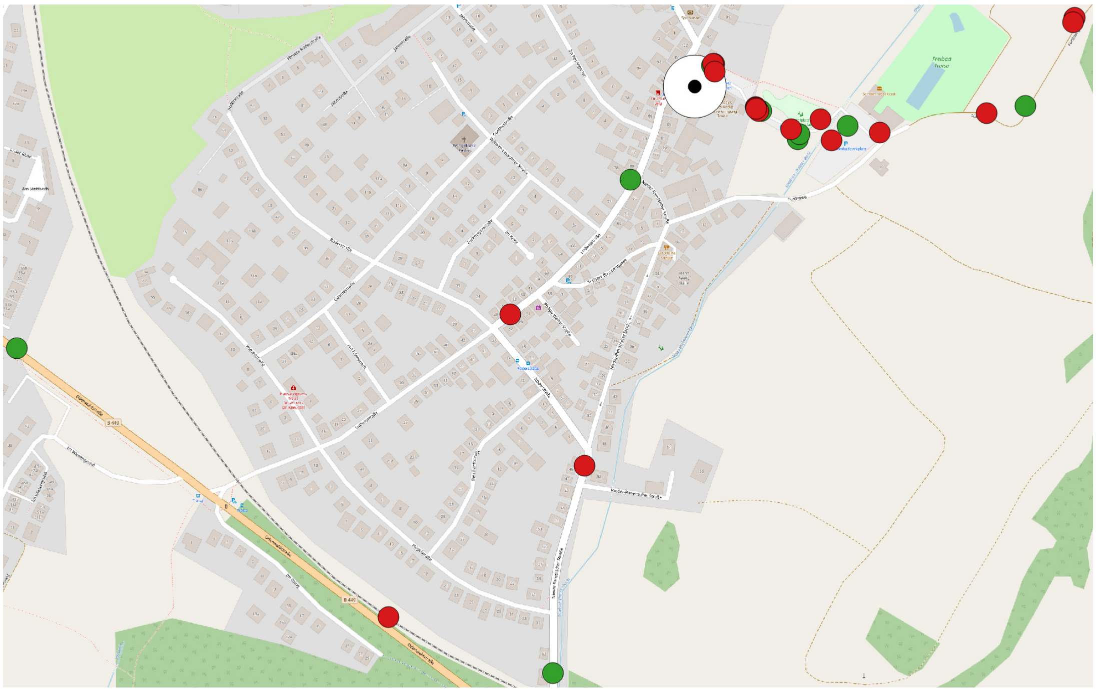


Figure 101: Indication of measurements with a successful (green) and unsuccessful (red) matching between the decoding results of ImdeaOWL and MobileInsight. The position of the serving cell tower is indicated by the black and white double circle in the north of the map. Map data ©OpenStreetMap contributors

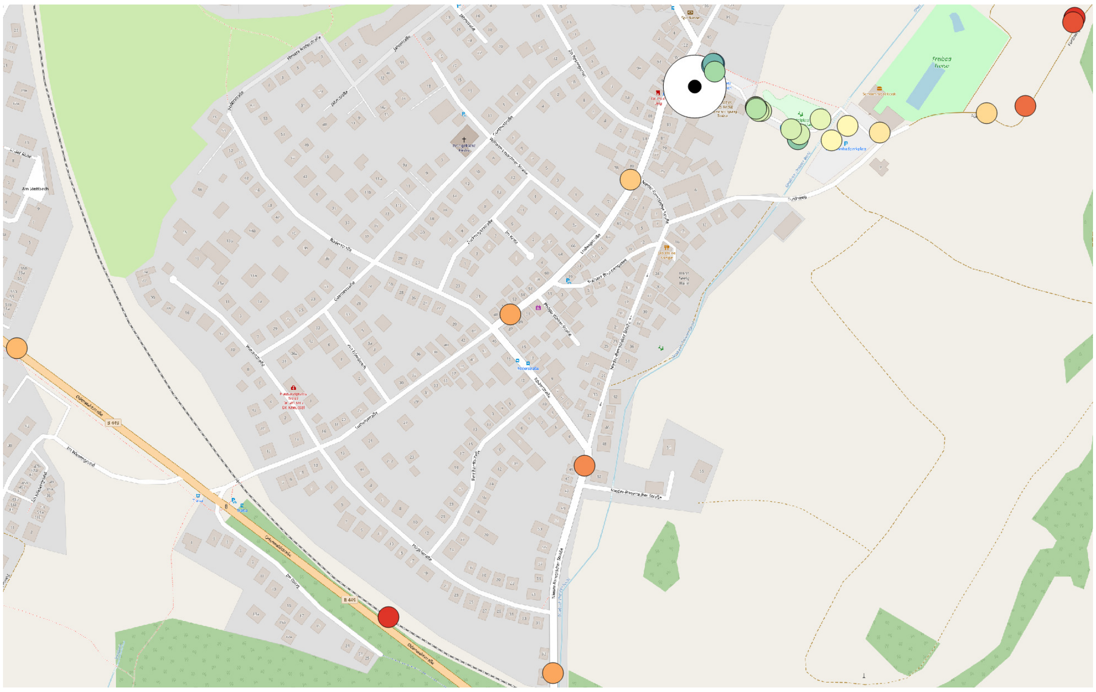


Figure 102: Visualization of the experienced Reference Signal Received Power (RSRP) values at the positions of our measurements. The RSRP values range from -108 dBm (red/poor) to -61 dBm (green/good). Map data ©OpenStreetMap contributors

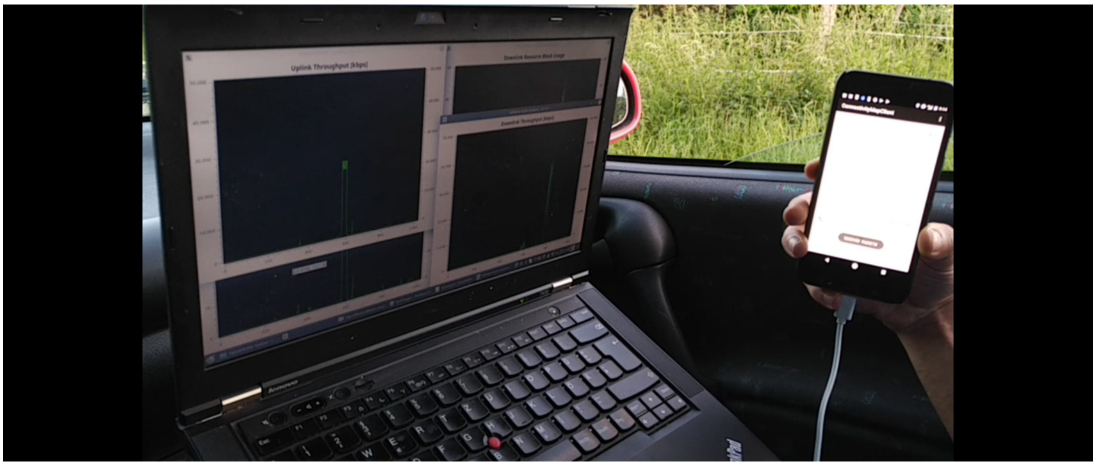


Figure 103: Setup of the ImdeaOWL network sniffer running on a laptop interfacing with a bladeRF SDR with its receiving antenna mounted on the car roof. The laptop is connected via USB to our test smartphone running an adapted version of our Connectivity Map Client application to receive signal, location and frequency readings to evaluate and trigger the decoding of the LTE network channel via ImdeaOWL. Each time the smartphone executes a test measurement, Imdea OWL is triggered to decode the LTE control channel.

A.16 LIST OF ACRONYMS

BASt	German Federal Road Research Institute (Bundesanstalt Für Straßenwesen)
CAM	Cooperative Awareness Message
CPU	Central Processing Unit
CQI	Channel Quality Indicator
C-RNTI	Cell Radio Network Temporary Identity
DENM	Decentralized Environmental Notification Message
EARFCN	Evolved-UTRA Absolute Radio Frequency No
EDGE	Enhanced Data Rates For GSM Evolution
ETSI	European Telecommunications Standards Institute
E-UTRA	Evolved UMTS Terrestrial Radio Access
GNSS	Global Navigation Satellite Systems
GPS	Global Positioning System
GUI	Graphical User Interface
HD Map	High Definition Map
HSPA	High Speed Packet Access
HTTP	Hypertext Transfer Protocol
ICCOMQS	Intelligent Cellular Communication Quality Sensing Framework
ISED	Canadian Organisation Of Innovation Science And Economic Development
Ko-HAF	Cooperative Highly Automated Driving (Kooperatives Hochautomatisiertes Fahren)
LDM	Local Dynamic Map
LTE	Long Term Evolution
MCS	Modulation And Coding Scheme
MIMO	Multiple Input Multiple Output
mMTC	Massive Machine-type Communications
MTU	Maximum Transmission Unit
NNI	Nearest Neighbour Index
OFDMA	Orthogonal Frequency Division Multiple Access
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase-Shift Keying
QUIC	Quick UDP Internet Connections
QXDM	QUALCOMM's EXtensible Monitor Software

RSRP	Reference Signal Received Power
RSRQ	Reference Signal Received Quality
RSSI	Received Signal Strength Indicaton
SAE	Society Of Automotive Engineers
SC-FDMA	Single Carrier Frequency Division Multiple Access
SINR	Signal To Noise Ratio
SISO	Single Input Single Output
TBS	Transport Block Size
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
uMTC	Ultra-reliable Machine-type Communications
UMTS	Universal Mobile Telecommunications System
VDA	German Association Of The Automotive Industry (Verband Der Automobilindustrie)
WiFi	Wireless Fidelity
WLAN	Wireless Local Area Network
xMBB	Extreme Mobile Broadband

A.17 SUPERVISED STUDENT THESES

[1] Samet Varol. "Algorithmic quality evaluation of the mobile radio network in the automotive application scenario." Bachelor Thesis. Hochschule Darmstadt, 2017.

[2] Ian Patrick Friedrich. "Aggregation and Prediction of LTE Network Quality with NS3." Bachelor Thesis. TU Darmstadt, 2017.

[3] Matthias Heinrich. "Map change detection in a high definition street map for highly automated driving vehicles." Bachelor Thesis. TU Darmstadt, 2017.

[4] Helmi Omrane Ben Sassi. "Network usage profiling for vehicular communication - Investigation of the usability of a low-cost LTE control channel decoder in the automotive context." Bachelor Thesis. TU Darmstadt, 2018.

[5] Pratik Kumar Mankar. "Accurate Map Creation from Mobile Sensor Data." Master Thesis. TU Darmstadt, 2016.

[6] Aakash Sharma. "Differential Map Updates for Highly Automated Driving and Enhanced Driver Assistance Systems." Master Thesis. TU Darmstadt, 2016.

[7] Rainer Wahler. "Analysis of Application-level Throughput and Load-balancing within an operational LTE Network using Smartphones." Master Thesis. TU Darmstadt, 2017.

[8] Markus Grau. "Development of intelligent approaches to estimate the quality of cellular networks with timely tolerant measuring data." Master Thesis. TU Darmstadt, 2017.

[9] Alexander Herzberger. "Prediction of data throughput in the LTE mobile radio network with machine learning methods in the automotive environment." Master Thesis [received the Alfred Trossen Price for the best master thesis of the year in the department of telecommunications engineering]. TH Bingen, 2017.

[10] Balpreet Jaggi. "Adaptive Throughput Measurement of the Cellular Networks: Investigating the measurement parameters to find an optimum trade-off between accuracy and cost." Master Thesis. TU Darmstadt, 2018.

[11] Florian Fischer. "Enhancement of mobile measurements of the cellular network for realization of a robust vehicular communication through status messages of the LTE protocol stack." Master Thesis. TU Darmstadt, 2018.

[12] Haroon Khaleeq. "Ad-hoc sharing of high defintion street map material for highly automated vehicles." Master Thesis. TU Darmstadt, 2018.

B

AUTHOR’S PUBLICATIONS

MAIN PUBLICATIONS

[1] Florian Jomrich, Daniel Bischoff, Steffen Knapp, Tobias Meuser, Björn Richerzhagen, and Ralf Steinmetz. “Lane Accurate Detection of Map Changes based on Low Cost Smartphone Data.” English. In: *In Proceedings of the 5th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS)*. Vol. 5. May 2019, p. 12.

[2] Florian Jomrich, Josef Schmid, Steffen Knapp, Alfred Höß, Ralf Steinmetz, and Björn Schuller. “Analysing communication requirements for crowd sourced backend generation of HD Maps used in automated driving.” In: *Proceedings of the 2018 IEEE Vehicular Networking Conference (VNC)*. Dec. 2018, p. 8.

[3] Florian Jomrich, Florian Fischer, Steffen Knapp, Tobias Meuser, Björn Richerzhagen, and Ralf Steinmetz. “Enhanced Cellular Bandwidth Prediction for Highly Automated Driving.” In: *Smart Cities, Green Technologies, and Intelligent Transport Systems 7th International Conference, SMARTGREENS 2018, and Fourth International Conference, VEHITS 2018, Funchal, Madeira, Portugal, March 16-18, 2018, Revised Selected Papers*. Ed. by Brian Donnellan et al. Springer, Dec. 2019. ISBN: 978-3-030-26633-2.

[4] Florian Jomrich, Alexander Herzberger, Tobias Meuser, Björn Richerzhagen, Ralf Steinmetz, and Cornelius Wille. “Cellular Bandwidth Prediction for Highly Automated Driving - Evaluation of Machine Learning Approaches based on Real-World Data.” In: *Proceedings of the 4th International Conference on Vehicle Technology and Intelligent Transport Systems 2018*. Ed. by Markus Helfert and Oleg Gusikhin. 4. SCITEPRESS, Mar. 2018, pp. 121–131. ISBN: 978-989-758-293-6.

[5] Florian Jomrich, Markus Grau, Tobias Meuser, The An Binh Nguyen, Doreen Böhnstedt, and Ralf Steinmetz. “ICCOMQS - Intelligent measuring framework to ensure reliable communication for highly automated vehicles.” In: *Proceedings of 2017 IEEE Vehicular Networking Conference (VNC)*. IEEE, Nov. 2017, pp. 311–318. ISBN: 978-1-5386-0985-9.

[6] Florian Jomrich, Aakash Sharma, Tobias Rückelt, Doreen Böhnstedt, and Ralf Steinmetz. “Intelligent Offloading Distribution of High Definition Street Maps for Highly Automated Vehicles.” In: *Smart Cities, Green Technologies, and Intelligent Transport Systems 6th International Conference, SMARTGREENS 2017, and Third International Conference, VEHITS 2017, Porto, Portugal, April 22-24, 2017, Revised Selected Papers*. Ed. by Brian Donnellan et al. Springer, Aug. 2017. ISBN: 978-3-030-02906-7.

[7] Florian Jomrich, Aakash Sharma, Tobias Rückelt, Daniel Burgstahler, and Doreen Böhnstedt. "Dynamic Map Update Protocol for Highly Automated Driving Vehicles." English. In: *Proceedings of the 3rd International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2017)*. Ed. by Oleg Gusikhin, Markus Helfert, and António Pascoal. Vol. 3. Full Paper. SCITEPRESS – Science and Technology Publications, Lda., Apr. 2017, pp. 68–78. ISBN: 978-989-758-242-4.

[8] Florian Jomrich, Tobias Rückelt, Doreen Böhnstedt, and Ralf Steinmetz. "An Efficient Heat-Map-Based Wireless Communication Simulation Model for Omnet++." English. In: *Proceedings of the AmE 2017 - Automotive meets Electronics*. Vol. 8. Berlin · Offenbach, Bismarckstraße 33, 10625 Berlin: VDE Verlag GmbH, Mar. 2017, pp. 65–70. ISBN: 978-3-8007-4369-8.

[9] Florian Jomrich. "Intelligent Provisioning of High Definition Street Maps for Highly Automated Driving Vehicles." English. In: *Proceedings of: International Conference on Networked Systems (NetSys 2017)*. Mar. 2017, p. 2. URL: `netsys17.uni-goettingen.de/wp-content/uploads/2017/03/phdforum-paper15.pdf`.

[10] Florian Jomrich. *Nahtloser Handover in drahtlosen Fahrzeug-Kommunikationsnetzen*. Deutsch, German. Ed. by BestMasters. Springer Vieweg, Jan. 2016. ISBN: 978-3-658-13301-6.

CO-AUTHORED PUBLICATIONS

[11] Mathias Schneider, Josef Schmidt, Alfred Höß, and Florian Jomrich. "Vehicle to Server Communication for Highly Automated Driving in the Ko-HAF Project." In: *Application-Oriented Higher Education Research (AOHER)* (Mar. 2019), p. 13. ISSN: 2096-2045.

[12] Tobias Rückelt, Daniel Burgstahler, Florian Jomrich, Doreen Böhnstedt, and Ralf Steinmetz. "Impact of Time in Network Selection for Mobile Nodes." In: *In Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM 16)*. ACM Press. ACM, Nov. 2016, pp. 68–77. ISBN: 978-1-4503-4502-6.

[13] Tobias Rückelt, Florian Jomrich, Daniel Burgstahler, Doreen Böhnstedt, and Ralf Steinmetz. "Publish-Subscribe-Based Control Mechanism for Scheduling Integration in Mobile IPv6." In: *In: Proceedings of the 40th IEEE Conference on Local Computer Networks (LCN)*. Ed. by IEEE. IEEE, Oct. 2015, pp. 478 –481. ISBN: 978-1-4673-6771-4.

[14] Fabian Kaup, Florian Jomrich, and David Hausheer. "Demonstration of NetworkCoverage – A Mobile Network Performance Measurement App." In: *International Conference on Networked Systems (NetSys)*. Vol. 77. Mar. 2015, pp. 228–230.

[15] Mojisola Erdt, Florian Jomrich, Katja Schüler, and Christoph Rensing. "Investigating Crowdsourcing as an Evaluation Method for TEL Recommenders." en. In: *Proceedings of ECTEL meets ECSCW 2013, the Workshop on Collaborative Technologies for Working and Learning*. Ed. by Monica Divitini, Tobias Ley, Stefanie Lindstaedt, Viktoria Pammer, and Michael Prilla. Vol. 1047. http://ceur-ws.org: CEUR Workshop Proceeding Series, Sept. 2013, pp. 25–29.

DEMO PAPERS

[16] Florian Jomrich, Tushar Wankhede, Tobias Rückelt, Daniel Burgsthaler, Doreen Böhnstedt, and Ralf Steinmetz. "Demo: Rapid Cellular Network Simulation Framework for Automotive Scenarios (RACE Framework)." English. In: *Proceedings of: International Conference on Networked Systems (NetSys 2017)*. IEEE, Mar. 2017, pp. 1–2. ISBN: 978-1-5090-4395-8.

C

CURRICULUM VITÆ

PERSONAL

Name	Florian Jomrich
Date of Birth	17th January 1990
Place of Birth	Rüsselsheim am Main
Nationality	German

EDUCATION

Since 09/2015	Technische Universität Darmstadt, Germany Doctoral candidate at the Department of Electrical Engineering and Information Technology
09/2013-06/2015	Technische Universität Darmstadt, Germany Master studies in Informationsystemtechnologie (Informationssystemtechnik) Degree: Master of Science
04/2010-08/2013	Technische Universität Darmstadt, Germany Bachelor studies in Informationsystemtechnologie (Informationssystemtechnik) Degree: Bachelor of Science
07/2009-03/2010	Haus der Senioren, Rüsselsheim, Compulsory Community Service - Zivildienst
08/2000-06/2009	Max Planck Gymnasium, Degree: Allgemeine Hochschulreife

ACADEMIC EXPERIENCE

09/2015-12/2018	Researcher funded by the Opel Automobile GmbH and the BMWI (Federal Ministry of Economics and Energy) in the Collaborative Research Project Ko-HAF ”Cooperative Highly Automated Driving”

WORK EXPERIENCE

Since 09/2015	Technische Universität Darmstadt, Research assistant at the research group

	Distributed Sensing Systems at the Multimedia Communications Lab (KOM)
09/2015-06/2019	As a PhD Candidate at the Opel Automobile GmbH in Rüsselsheim, Germany, at the department of Electrical Engineering Advanced Safety and Controls, working on the German Ministry of Economics founded research project Ko-HAF (Cooperative Highly Automated Driving)

TEACHING ACTIVITY

since 2016	Lab and Project "Multimedia Communications Lab/Project", Organizer and Supervisor..

HONORS

05/2019	Best Paper Award for "Lane Accurate Detection of Map Changes based on Low Cost Smartphone Data" at VEHITS 2019.
2018	Supervision of the best Masterthesis of the year 2018 of the Department of Telecommunications Engineering of the TH Bingen by Alexander Herzberger for "Prediction of the throughput of LTE networks in the vehicular environment using machine learning techniques" honored with the Alfred-Trossen-Price.

Darmstadt, August 6th, 2019

Florian Jomrich

D

ERKLÄRUNG LAUT PROMOTIONSORDNUNG

---

§ 8 Abs. 1 lit. c PromO
Ich versichere hiermit, dass die elektronische Version meiner Dissertation mit der schriftlichen Version übereinstimmt.

§ 8 Abs. 1 lit. d PromO
Ich versichere hiermit, dass zu einem vorherigen Zeitpunkt noch keine Promotion versucht wurde. In diesem Fall sind nähere Angaben über Zeitpunkt, Hochschule, Dissertationsthema und Ergebnis dieses Versuchs mitzuteilen.

§ 9 Abs. 1 PromO
Ich versichere hiermit, dass die vorliegende Dissertation selbstständig und nur unter Verwendung der angegebenen Quellen verfasst wurde.

§ 9 Abs. 2 PromO
Die Arbeit hat bisher noch nicht zu Prüfungszwecken gedient.

Darmstadt, 06. August 2019

Florian Jomrich