DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Automotive Software Engineering

Route Determination for Teleoperated Driving based on Cellular Network Availability

Christoph Marc Weinisch

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Automotive Software Engineering

Route Determination for Teleoperated Driving based on Cellular Network Availability

Routenbestimmung für Teleoperiertes Fahren auf Basis der Verfügbarkeit des Mobilfunknetzes

Author:	Christoph Marc Weinisch
Supervisor:	Prof. Dr.-Ing. Jörg Ott
Advisor:	Stefan Neumeier
Submission Date:	15.04.2019

I confirm that this master's thesis in automotive software engineering is my own work and I have documented all sources and material used.

Munich, 15.04.2019 Christoph Marc Weinisch

Abstract

Due to ever newer driver assistance systems and better sensor technology, it is only a matter of time before autonomous driving will prevail. However, there are still some barriers to overcome, such as legal questions or what happens in the event of a sensors failure. For this reason, teleoperated driving can serve as a transitional technology, alternative or supplement to autonomous driving. Via wireless communication, it is possible to send a live video stream from the vehicle and a teleoperator can control the vehicle via appropriate control signals from his operator workstation. Within this work, an overview of the development of the different network standards is given. As teleoperated driving and the associated necessary video stream require high data rates and control signals in real time, the existing LTE infrastructure is used in this work as the basis for the provision of a mobile radio connection. It is of enormous importance that the routes are defined before the start of the journey, to ensure that a route does not lead through road sections in which there is no stable network connection. Therefore the routes are determined depending on the availability of the LTE network. In this thesis, two different approaches are designed to project the availability of the LTE network on a road network and then to evaluate the routes. The network simulation will be implemented with the help of OMNeT++ and the vehicle traffic simulation with the help of SUMO. Veins is the used interface, which provides such a scenario for Erlangen. On the basis of this example scenario, a mathematical approach is implemented which determines the availability of the network on the basis of the approximated transmission range of the stationary transmission units. A second approach involves a simulation to block roads from the road network where no LTE network availability was determined. For this purpose, measuring vehicles are simulated that travel the entire road network in order to check the location-dependent availability of the network. The resulting road networks are then compared with the unprocessed road network using a vehicle simulation.

Kurzfassung

Durch immer neuere Fahrerassistenzsysteme und bessere Sensorik ist es nur noch eine Frage der Zeit, bis sich das autonome Fahren durchsetzen wird. Allerdings gibt es noch einige Widrigkeiten zu überwinden, wie zum Beispiel die rechtliche Situation oder was bei einem Komplettausfall der Sensorik passiert. Aus diesem Grund kann das teleoperierte Fahren als Übergangstechnologie, Alternative oder Ergänzung zum autonomen Fahren dienen. Über eine drahtlose Kommunikation ist es möglich einen Live-Video-Stream aus dem Fahrzeug zu senden und ein Teleoperator kann von seinem Operator-Arbeitsplatz das Fahrzeug steuern. Hierzu wird ein Überblick über die Entwicklung der verschiedenen Netzstandards gegeben. Da für teleoperiertes Fahren hohe Datenraten und Steuersignale in Echtzeit erfolgen müssen, wird in dieser Arbeit die bereits vorhandene LTE Infrastruktur als Basis für die Bereitstellung einer Mobilfunkverbindung genutzt. Wichtigist hierbei, dass die Routen vor Fahrtantritt definiert werden, sodass eine Route nicht Bereiche ohne Netzabdeckung führt. Das heißt, dass die Routen in Abhängigkeit der Verfügbarkeit des LTE-Netztes bestimmt werden. Hierzu werden in dieser Arbeit zwei Ansätze konzipiert um die Verfügbarkeit des LTE-Netztes auf ein Straßennetz zu projektieren und anschließend die Routen zu evaluieren. Die Netzwerksimulation wird dabei mit OMNeT++ und die Fahrzeugverkehrssimulation mit SUMO umgesetzt. Als Schnittstelle agiert dabei Veins, welches ein solches Szenario für Erlangen bereitstellt. Anhand dieses Beispiel-Szenarios wird einmal ein mathematischer Ansatz umgesetzt, welcher die Verfügbarkeit des Netzes anhand von den approximiertem Senderadien der ortsfesten Sendeeinheiten festlegt. Ein zweiter Ansatz führt eine Simulation durch, um die Verfügbarkeit des LTE-Netzes in dem ausgewählten Ausschnitt des Straßennetzes abzubilden. Dafür werde Messfahrzeuge simuliert, die das gesamte Straßennetz abfahren, um die ortsabhängige Verfügbarkeit des Netzes zu überprüfen. Dadurch entstehen zwei verschiedene Straßennetzwerke, welche dann mit dem unbearbeiteten Straßennetz mithilfe einer Fahrzeugsimulation miteinander verglichen werden.

Contents

1 Introduction

In this section, the motivation for the topic of this work will be discussed first. Subsequently, the research question that guides this thesis will be formulated and subdivided into subitems. This research question will be answered at the end of the thesis in Chapter 6. Furthermore, the goals and application possibilities will be presented and finally the structure of this work will be outlined.

1.1 Motivation

Autonomous driving will definitely become established over the next few years or decades and the existing difficulties will gradually be solved. Nowadays, only a few highway scenarios and traffic jams or stop-and-go traffic partly fulfil the necessary prerequisites. For all other scenarios and above all for urban traffic, it can definitely be said that no autonomous vehicles will be able to move on public roads in the coming years because the complexity, with many different road users and confusing road topologies, is far too great. Furthermore, the legal situation also prevents the use of autonomous vehicles. For this reason, research is being conducted into teleoperating and semi-autonomous control of vehicles. This concept, borrowed from robotics, brings a human driver back into the control loop in order to master complexity. This procedure is already being used in deep-sea or space research. The operator is connected to the vehicle by using the cellular network.

In order to control cars with this concept, the routes must be determined before starting the route. In particular, the availability of the network at any time on the route must be taken into account, otherwise the control of the vehicle is no longer possible. The objective is to investigate to what extent these routes deviate from "normal" routes and how large changes to efficient routes are necessary in order to find suitable routes for teleoperated driving. Various approaches can be tested and compared to determine the most efficient routes for teleoperated driving. The route determination must be completed before the start of the journey in order to ensure that there is no network interruption at any time during the trip.

1.2 Research Questions

How can cellular network availability be projected to a digital road network to enable route determination for teleoperated driving?

The progress of technology is constantly increasing and with it also the automation of vehicles up to autonomous vehicles. Since the status quo is not yet ready to completely replace a human being, a vehicle could alternatively be maneuvered teleoperatively [1]. The conventional route determination in the road network is not designed for a teleoperated system, as these do not have any information about the availability of the vehicle's connection to the remote driver. A routing through an area without cellular network coverage could have serious consequences. Therefore an alternative to the conventional route determination is necessary.

1.3 Outline of the work

The following Chapter 2 describes the general information about this work. First the autonomous driving is presented, what the technology and the current state in this area is and explained then in the next section, why teleoperated driving can help to advance this technology. Furthermore, this chapter explains the different mobile network standards. The focus here is on the LTE network, which is the decisive network standard for this work. The tools used and general standards for route determination are also described and explained here.

In Chapter 3 the methodologies of the the Mathematical Approach and the Simulative Approach are described, which are compared during this thesis.

Chapter 4 takes the two approaches up and explains how they have been implemented in the practical part of this paper. The first part "Environment" explains how the real part of the Erlangen road network is converted into a SUMO compatible road network. First, the implementation of the mathematical approach is described and the resulting road network is demonstrated. Subsequently, the individual steps for the implementation of the simulative approach are described and the resulting road network then described.

In Chapter 5 the two approaches and the resulting road networks are compared and the effects on route determination are illustrated.

In the last Chapter 6 the research questions are taken up again and answered. A summary of the whole work is given and topics are discussed which are of interest for future work.

2 Related Background

Within this chapter, we first discuss the basics and current status of autonomous driving and provide some background knowledge of the tools used for the simulation. Secondly, we describe the fundamentals of teleoperated driving and explain the vital connection between teleoperated driving and autonomous driving. Thirdly, we define various mobile radio standards starting with a general overview and the development history, the mobile radio standards GSM, GPRS and EDGE and UMTS and HSPA. Furthermore, we describe the mobile radio standards LTE and 5G and explain the reasons why these mobile radio standards with high data rates are indispensable for teleoperated driving. Fourthly, we describe the tools used to realize the simulation with their main characteristics and why these were used in this thesis. Lastly, we give an overview of fundamental route determination problems. determination problems.

2.1 Autonomous Driving

Autonomous driving is the movement of vehicles that behave mainly autonomously. This involves various real-time systems working together which consist of different driver assistance systems and sensors, which are already standard in many vehicles today. In this section, the basic concepts of driver assistance systems and the necessary sensors in the context for autonomous driving are described. Furthermore, this section describes the current state of development and the various autonomy levels for autonomous driving.

Nowadays advanced driver assistance systems (ADAS) have become indispensable in new vehicles. The ADAS are becoming increasingly complex as they are no longer only passive, but also active systems interacting with the driver. These systems are electronic auxiliary equipment installed in vehicles in order to assist the driver in certain driving situations. Adaptive Cruise Control is an example of such a driver assistance system which has become a standard system in many premium car models. Another popular driver assistance system in motor vehicles is the Distance Cruise Control; this system measures the gap to the vehicles in front, uses this measurement as additional feedback and manipulates this variable for vehicle control purposes (e.g. decelerating the vehicle). Other examples of such systems are active brake systems and lane departure warning systems.

If several driver assistance systems interact and thus the driver is no longer required to intervene, this is considered to be autonomous driving. Daniel Watzenig and Martin Horn describe several advantages that can be achieved by autonomous driving [2]:

- Improve safety by reducing human driving errors
- Significantly contribute to the optimization of traffic flow
- Help to reduce fuel consumption and CO2 emissions
- Enhance the mobility of seniors and unconfident drivers

Depending on the source, various forecasts state that high- or fully-automated driving functions will already exist by around 2040 [3]. These different levels of autonomy are described in section 2.1.1. The driver assistance systems which are already in use today form the basis for completely autonomous vehicles' design and production [2].

The top goal for automated driving includes Vehicle to Vehicle Communication (V2V) and Vehicle to Infrastructure Communication (V2I). These communicative channels are supposed to achieve the goals of automated driving as described above. V2V communication enables vehicles to transfer local traffic data, such as traffic accidents or congestion, to other vehicles nearby or on the way towards them. Furthermore, the driving intention of a vehicle can be communicated to the surrounding vehicles, thus improving the traffic flow and reducing the accident rate [2]. The V2I is used in automated driving to optimize the use of the road network, thereby reducing high traffic volumes and thus reducing environmental pollution. The role allocation between human drivers and automated driving systems is specified.

2.1.1 Levels of Autonomy

In Europe and the USA, the classification of autonomous driving is conducted in six stages which can be seen in table 2.1. In SAE Level 0 there is no automation at all. All aspects of the dynamic driving task are performed continuously by the human driver. These are the vehicles which are widely spread and the driver is responsible for the complete operation of the vehicle. Automated systems issues warnings and may momentarily intervene, but have no sustained vehicle control.

SAE Level	Name	Execution of Steering and Acceleration/Deceleration	Monitor	Fallback Performance
0	No automation	Human	Human	Human Driver
1	Driver assistance	Human / System	Human	Human Driver
2	Partial automation	System	Human	Human Driver
3	Conditional automation	System	System	Human Driver
4	High automation	System	System	System
5	Full automation	System	System	System

Table 2.1: Levels of automation

In SAE Level 1, the system takes over the tasks of driving mode-specific functions, whereby either the steering or the acceleration/deceleration is carried out by the system. This entails that the driver and the automated system share control of the vehicle. The driver is still in control of the overall operation and the safety of the vehicle. However, the car can take over at least one vital function for a finite period of time, but has no sustained vehicle control. A relevant example is the Adaptive Cruise Control (ACC), where the driver controls steering and the automated system controls the speed. Further examples are Parking Assistance systems and Lane Keeping Assistance (LKA). As can be seen in table 2.1, the system either takes control of the steering or is responsible of the acceleration/deceleration.

In SAE Level 2, driving mode specific tasks are performed by the system. Steering and acceleration/deceleration are performed by one or more driver assistance systems. In this level - partial automation - the driver can take his hands off the wheel. The driver is still responsible for the safe operation of the vehicle, but under certain conditions, the car can take over the steering, braking and acceleration functions. The big difference to SAE level 3 is that the driver is still responsible for monitoring the driving task which causes that the driver must be prepared at all times to take over the driving task immediately. Examples of this SAE level are the Mercedes-Benz Drive Pilot or the Tesla Autopilot.

In SAE Level 3, all driving mode-specific tasks are performed by an automated driving system that takes into account all aspects of the dynamic driving task. This means that the driver can withdraw his attention from the driving task. The vehicle can drive on its own and will handle situations that require an immediate response, such as emergency braking. The driver must also be prepared to intervene within a time limit set by the manufacturer when prompted to do so by the vehicle. Car manufacturers generally consider this to be the most dangerous level, as individuals often place too much trust in the technology and stop paying attention. An example of this SAE level is given by the Audi Traffic Jam Pilot (A8 Sedan) [4].

SAE level 4 can be seen as SAE level 3, but no driver attention is ever required for safety. Also in this case, all driving mode-specific tasks are performed by an automated driving system that takes into account all aspects of the dynamic driving task. That means the driver may safely go to sleep or leave the driver's seat. The car can drive itself full-time under the right circumstances and the car is also expected to have backup systems so that if one technology fails, it will still be operational. Outside the right circumstances, the vehicle must be able to safely abort the trip, for example by parking the car until the driver retakes the control and leads the vehicle back to a safe environment in which the systems can take over the driving task again.

SAE level 5 describes a fully automate driving task, meaning the whole driving process can be done by itself. It is a high performance, full-time automated driving system which takes over all the aspects of the dynamic driving task (that would be controlled by a human driver in a non-automated system), under all road and environmental conditions. That means that the car controls itself under all possible circumstances with no expectation of human intervention. These cars don't need steering wheels, accelerators or brake pedals and will likely look quite differently than the cars we see on the road today. An example of such a car would be a robotic taxi.

2.1.2 Sensors

The intervention and the signaling functions of driver assistance systems require knowledge of the current driving situation. In the case of ESP (Electronic Stability Program) these are wheel speed, the yaw rate, as well as longitudinal and lateral acceleration. New driver assistance systems such as ACC require additional information about the vehicle environment. Various types of environment sensors are used for this purpose, such as:

- Ultrasonic: Used for Parking Assistance and Distance Measurements in the short range (between 25 cm and 150 cm).

- Radar: Used for Lane Change Assistant and Automatic Distance Warning System. Two different frequency bands are distinguished here:

  - 24 GHz bandwidth: Short-range segment from 0.5 m and mid-range segment up to 100 m.
  - 77 GHz bandwidth: Long range segment up to 400 m.

- Lidar: Used for Blind angle monitoring, Automatic distance warning, Distance control, Pre-crash and Pre-brake systems. It is similar to radar, but instead of microwaves, ultraviolet, infrared or visible rays are emitted.
- Camera: Used for Lane departure warning, Traffic sign recognition, Lane change assistant, Blind spot monitoring, Emergency braking system for pedestrian protection.

2.1.3 Status Quo

Autonomous driving is nowadays already possible from the state of the art. However, this only applies in certain, less complex situations. These include some highway scenarios and congestion or stop-and-go traffic that partly meet these conditions. For many other scenarios, which are mainly in the area of urban traffic, one can definitely say that in the next few years no autonomous vehicles will be able to move on public roads. This is because the complexity, with many different road users and confusing road topologies, is far too great.

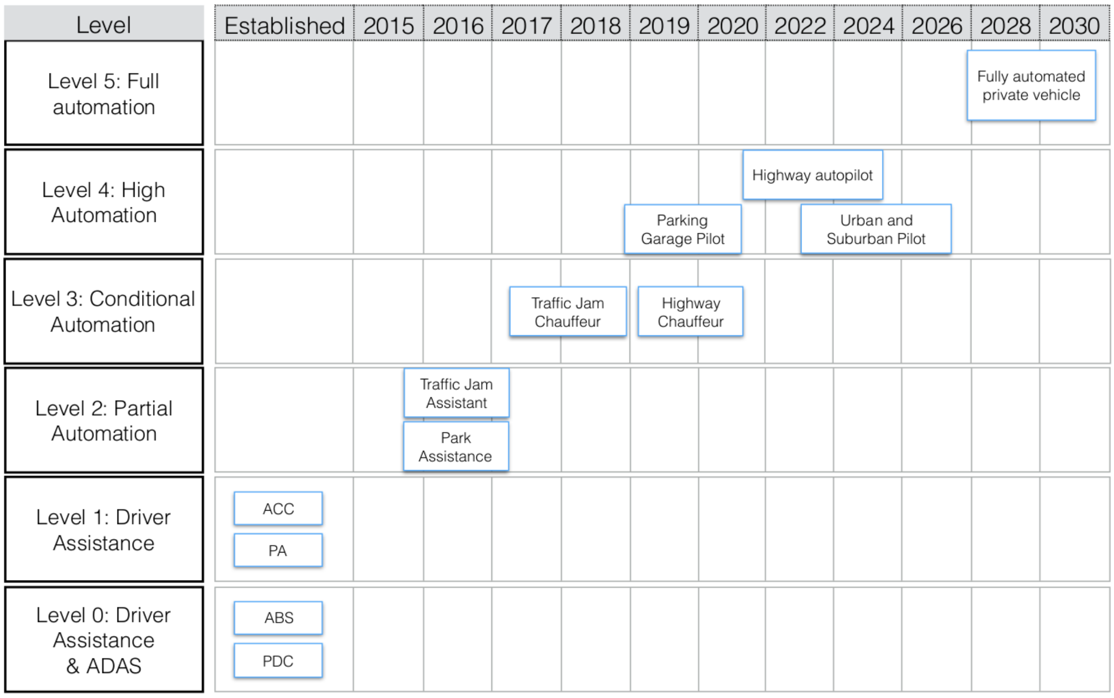

Figure 2.1: Connectivity and automated driving [1].

Most today's cars are already equipped with Level 1. Some modern cars already provide some Level 2 features and in particular cases even Level 3 features. From automation level 3, the system takes over the monitoring of the environment in predefined cases. According to the ERTRAC Task Force "Connectivity and Automated Driving" starting from approximately 2030 fully automated vehicles exist which correspond to the SAE level 5 as can be seen in Figure 2.1.

The automotive industry has made great progress in recent years and has integrated driver assistance systems that can perform up to SAE Level 3 tasks into series production automobiles. For the assumption of such challenging driving tasks, an extremely large amount of data of the vehicle's environment is required to calculate the driving task, transmitted by the various sensors described in the previous section. The sensor technology used and the corresponding processing of the receiving data is already used in today's standard driver assistance systems and thus continuously improved. The ADAS used today are therefore the basis for autonomous driving and this is one reason why from a purely technical point of view there is not a great lack to produce autonomous vehicles in series. The leading companies in this area are currently Waymo (Google) and Cruise (General Motors). But also the German automobile manufacturers are already very far advanced with the development of the new technology and it is a real competition for the time, which company will bring first a completely autonomous vehicle on the market.

Nevertheless, there are a number of problems which cause time to pass before autonomous vehicles can assert their claim. The main problem here is the legal situation. The first steps have already been taken and a new law which regulates automated driving within a uniform scope became effective in Germany in June 2017, for example. However, this law only applies to autonomous driving, in which a driver is ready to take over at any time as a backup. According to the levels of autonomy, this is not level 5 (full automation), since such vehicles should also drive without a steering wheel and accelerator pedals or brake pedals completely without human monitoring. Until autonomous driving can come to series production on the market, the following difficulties, among others, have to be overcome:

- Create a legal basis that also defines responsibility in the case of an accident.
- Expansion of the infrastructure: Equipping roads with sensors to enable automated cars to cope with complicated situations at traffic lights or intersections.
- Protection against data misuse: Clarifying how data from people moving around vehicles, such as pedestrians, is to be processed.
- Create user acceptance to rely on the new technology.
- Behavior in the event of complete system failure, making automated onward travel impossible.

2.2 Teleoperated Driving

The difficulties mentioned will lead to the consequence that in the coming years it is not to be expected that there will be series of vehicles that fulfill all the criteria of SAE Level 5. In order to overcome these difficulties, there is the idea of teleoperated driving which is constantly being expedited. This means that vehicles are teleoperated and semi-autonomously controlled. Teleoperated driving in this context means that an external operator controls the vehicle with the help of a live video. The operator is connected to the vehicle via wireless data transmission - e.g. via WLAN or cellular network. For example, in the event of a complete sensor failure or other difficulties, an external operator or would be able to steer the vehicle and bring it back into a safe environment.

2.2.1 State of the art

Teleoperation was originally a robotics methodology and has been used there since the early 20th century. However, it is only since the 1970s that teleoperated systems have been used on a larger scale. In general, remote-controlled robots are used whenever the application is too complex for complete automation, but the task cannot or should not be performed by a human.

Remote-controlled vehicles are used in a wide range of fields. The largest application for teleoperated robots is currently remote-controlled underwater vehicles. They are used to explore ecosystems of water bodies, salvage operations or oil wells. Teleoperated aircraft, also known as Unmanned Aerial Vehicles (UAV), are mainly used for military purposes. They are deployed, for example, for the exploration of an area or for target recognition and marking in crisis areas [5].

Despite the fact that teleoperated vehicles are used in a wide variety of application scenarios, according to Winfield there are three characteristic elements that every teleoperated application features [6]:

- **Operator workstation:** The human operator controls the robot from a control station. Therefore, the interface usually consists of at least one display to visualize the status of the robot. When driving teleoperated, this display usually shows at least one camera perspective of an on-board camera. In addition, environmental information such as navigation maps or radar data are frequently displayed, as well as general sensor information about the robot itself. Furthermore, the operator workstation consists of an input device from which freely definable control signals are generated. This input device can consist of a conventional steering wheel and pedals for a road vehicle, but it can also be a Joystick or touch screen are possible alternatives [7].

- **Communication connection:** The communication between robot and operator can be done by wireless or wired data transmission. In the case of teleoperated driving vehicles on roads, of course, only the wireless connectivity is applied. The connection must be designed to transmit the necessary data between the robot and the operator with the least possible delay. Video streaming is usually the most critical aspect in terms of time delay and bandwidth [8].

- **Robot:** The robot itself is very application-specific, but it is always characterized by the fact that it must be equipped with appropriate sensor, actuator and data transmission systems that enable teleoperated control [5].

2.2.2 Concepts for teleoperated driving in road traffic

In contrast to the above-mentioned teleoperations, road traffic is much more complex, as it involves a greater variety of restrictions that make interaction with other traffic participants indispensable. The risk of an application taking place in uninhabited areas - as the examples describe above - is significantly lower than when driving on public roads, since it can be assumed in the field that interaction with other dynamic objects is not necessary and that only static objects have to be considered as obstacles.

Figure 2.2 shows a basic concept for teleoperated driving in road traffic. The operator workstation must be equipped in such a way that the operator has exactly the information available that he needs to control the vehicle as well as an on-board driver. The concept of the workstation is mainly based on the display of visual information and consists of a structure with several monitors on which the camera perspectives of the on-board video cameras are displayed live. In addition, further information is displayed in the video image to make driving easier. In parallel, information from the environmental sensors is visualized to assist the teleoperator in estimating the distance and speed of other road users. This workstation is shown at the bottom right in the figure 2.2. An auditory telepresence is particularly relevant for the feeling of speed, the recognition of driving conditions and the location of other road users. It is to be realized by means of a sourround sound system, since the ear is also important for driving vehicles as well [9].

Car-2-server communication is used as the communication structure. The vehicle is permanently connected via cellular network and Internet to a server, which provides data to support the driver. The teleoperated vehicle uses this existing Car-2-Server connection and transmits video and sensor data and receives control signals. Due to the high volume of data generated by live video streaming, next-generation mobile radio networks (4G) are an essential prerequisite for teleoperated driving. In addition, high standards must be set for availability and data encryption [5].

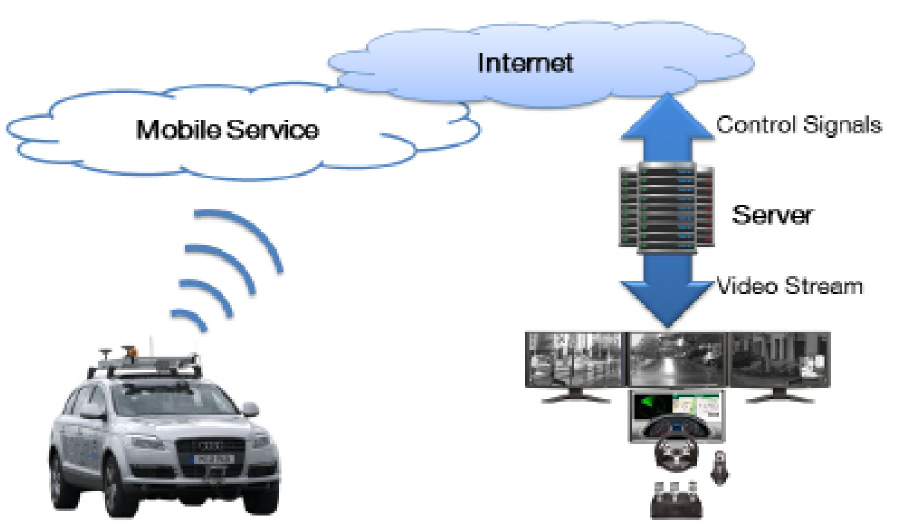

Figure 2.2: Data transmission scheme for teleoperated road vehicles [10].

.

In this case, the robot is a road vehicle in which all elements of the primary, secondary and tertiary vehicle guidance can be controlled electronically. In order to give the driver a similar view of the traffic situation as an on-board driver, a camera concept with several cameras should be used. Multiple cameras should provide a comprehensive 200 to 240 degree forward view. For rear view, cameras in the exterior mirrors and a rear view camera with back vision are provided. The concepts are designed for an application within mobility concepts in urban scenarios and were initially investigated for speeds of up to 50 km/h only. From the test drives in the simulation the results could be obtained that a total delay time of 400 to 500 milliseconds represents the maximum of the tolerable time delay [8][5].

2.2.3 Fields of application

The first test drives with teleoperated test vehicles have already shown that this methodology currently has clear advantages over purely autonomous vehicles. The operator is much better able to quickly understand and interpret even complex situations and, above all, to correctly anticipate the behavior of other road participants. If the situation can be controlled by machines, automated systems have advantages over humans in terms of reaction time and reliability. Teleoperated vehicles, on the other hand, can be used in predominantly complex traffic situations, since a human sensor can recognize every situation and drive the vehicle in a safe manner. There are a number of different application areas for teleoperated driving. Here are some typical examples for large cities that may occur in the near future [11]:

- Delivering and collecting Car-Sharing vehicles directly to the front door of the user's home.
- Parking services in city areas.
- Remote control of electric vehicles to bring them to the next charging station.

2.3 Cellular Network

The term Cellular Network refers to the technical infrastructure on which the transmission of signals for mobile communications takes place. The mobile network essentially comprises the mobile switching network, in which the transmission and switching of the signals between the fixed devices and platforms of the mobile network is carried out, as well as the access network, in which the transmission of the signals between a mobile antenna and the mobile phone is realized [12]. The oldest still operated network in Germany is the D-Net, which was introduced in 1991 and is based on the GSM standard. At that time, voice transmission was the most common form. In the mid-1990s, however, data transmission gained in importance and required the modernisation of GSM. To achieve this, the GPRS standard was developed and introduced. For this purpose, the mobile switching network was extended to enable packet-oriented data transmission. GPRS extends GSM by the capability of data transmission. Both systems use the same base stations, so that with the exception of a new software version no changes were made in the access network. However, the core network has been extended to include the ability to establish packet-switched communication links. This enables flexible bandwidths and thus improves the efficiency of the transmission capacity of the channels [13].

The introduction of data-optimized 3G networks aims at increasing data transmission rates. Extended functionalities, for example in the multimedia field, follow with the expansion of the 3G networks. The most important 3G standard is the Universal Mobile Telecommunications System (UMTS). The market introduction of smartphones, which increasingly support data applications, further increased the capacity requirements of the networks, leading to the development of the Long Term Evolution (LTE) standard. LTE only uses data transmission and is therefore different from older networks that also use circuit switching.

2.3.1 GSM

The Global System for Mobile Communications is a mobile communications standard introduced in 1990 for fully digital mobile networks. It is mainly used for telephony, but also for circuit-switched and packet-switched data transmission and text messages. It is the first standard of the so-called second generation (2G) as successor of the analog systems of the first generation.

In a circuit-switched digital telecommunications network which is explained in the next section, a voice channel typically occupies a 64 kbit/s timeslot. The PCM method is used to convert analog speech for digital transmission. The ranges that can be achieved with GSM vary greatly depending on the terrain profile and buildings. In the open air up to 35 km can be reached by visual contact. For longer distances, the signal propagation time of the radio signals prevents communication between the base station and the mobile station [12][14].

Network architecture GSM networks are divided into five subsystems:

- **Mobile Station (MS)**: The mobile station consist of an antenna to which a transmitter and receiver unit is connected, a power supply, a speaker and a microphone or an external connector and a possibility to select another subscriber (typically keyboard). Another essential component of MS is the SIM card.

- **Base Station Subsystem (BSS)**: The BSS consists of at least one base station (BTS, Base Transceiver Station), usually several (usually some 10 to some 100). Each base station operates one or more radio cells via the antennas connected to it. The base stations are connected to a central control unit (BSC, Base Station Controller), which monitors the connections and initiates cell changes (handovers) if necessary.

- **Network Switching Subsystem (NSS)**: The NSS consists of the MSC (Mobile Services Switching Centre), which is the actual exchange and interface between the radio network and the telephone network. The NSS also includes the VLR (Visitor Location Register), which stores information on all mobile subscribers within the radio network. The HLR (Home Location Register), on the other hand, stores information about all subscribers who are customers of the radio network owner. The AUC (Authentication Center) is responsible for authentication [12].

- **GPRS Core Network**: For the packet-switched part GPRS the SGSN (Serving GPRS Support Node) and GGSN (Gateway GPRS Support Node) are available [12].

- **Network Management Center (NMC)**: The NMC monitors the mobile network and controls the MSC, BSC and BTS [15].

2.3.2 GPRS and EDGE

General Packet Radio Service (GPRS)

GPRS extends GSM by the capability of data transmission. Both systems use the same base stations, so no changes have been made to the access network except for a new software version. However, the core network has been extended to include the ability to establish packet-switched communication links [16]. For this purpose, the Packet Control Unit (PCU) for reserving the timeslots in the base stations, the Serving GPRS Support Node (SGSN) for routing the data packets between the individual radio networks and the Gateway GPRS Support Node (GGSN) for connecting the mobile network to the Internet were added to the network [12]. If GPRS is activated, there is only a virtual permanent connection to the remote terminal. The radio room is only occupied when data is actually transmitted, otherwise it is free for other users. Therefore, no radio channel needs to be permanently reserved for one user.

Since the GSM network was originally conceived as a circuit-switched network, the exclusive channel is used for a conventional voice or data connection between two subscribers. During the connection, this channel can only be used by the two interconnected participants. In order to enable packet data transmission in GSM networks as well, the General Packet Radio Service (GPRS) was established. Special emphasis was placed on the fact that no new base stations (BTS's) are required for GPRS. This was an important prerequisite for cost-effective, packet-oriented transmission in already existing networks [12][17].

Circuit Switching

Circuit switching provides the basis for traditional telephone networks. If a call is made, a circuit switch creates a temporary and dedicated link of fixed bandwidth between communicating end nodes. This link only lasts until the call is complete. If there are not enough network resources available, the call will not be established or "completed as dialed". One of the biggest disadvantages of circuit switching, the big bandwidth wastage is circumvented by the introduction of the Packet Switching [12].

Packet Switching

Packet switching allows users to equally share bandwidth resources but makes no promises concerning quality or latency. This is useful for transferring data that do not require real-time responsiveness. Packet switching places the intelligence in the end nodes, rather than the phone company facilities, with a simple underlying network that only directs packets from one side to the other. Packet switching is easier and more affordable than circuit switching. Since all the bandwidth can be used at once, packet switching is more efficient because it does not have to deal with a limited number of connections that may not be using all that bandwidth. Packet switching also requires a less complicated infrastructure that can easily respond to network failures. This makes it easier and more cost-effective to add new nodes whenever they are needed [18].

GPRS is packet-oriented which means that the sender converts the data into individual packets, transmits them as such and reassembles them at the receiver. GPRS technology theoretically enables a data transmission rate of 171.2 kilobits per second for the bundling of all eight GSM time slots of a channel. In practical operation, however, the number of usable time slots within a frame is limited by the multislot capability of the mobile station and the networks. This transmission rate is only a theoretically achievable value. In practice, the network operators limit themselves to 53.6 kBit/s.

Since the year 2000 most mobile phones support GPRS. This was used for viewing WAP pages or for example for the Multimedia Messaging Service (MMS). There are three different classes of terminal devices that indicate the basic capabilities of a terminal device [12].

- **Class-A**: Class A terminal devices make it possible to use packet-switched and circuit-switched services simultaneously.
- **Class-B**: Class B terminal devices may use either a packet switched or a circuit switched service, but not both at the same time.
- **Class-C**: Class C devices only allow data transmission as long as they are registered for the GPRS service.

Enhanced Data Rates for GSM Evolution (EDGE)

In order to increase the GPRS transmission speed even further, GPRS has been defined as a next expansion stage a further modulation method according to the 8PSK principle under the name Enhanced Data Rates for GSM Evolution(EDGE). EDGE is a further development of the GSM technology. It is basically a standard GSM with more bits per symbol, i.e. a higher bit rate with the same symbol rate. Many of the originally used components in the radio network were not compatible with the new modulation method. Nevertheless, EDGE could be integrated into mobile radio networks with moderate effort, since it does not interfere with existing mobile telephony. Essentially, it is necessary to update the software of the GSM base station and replace individual components if necessary. The increase of the data transfer rate in EDGE is achieved by switching to a more efficient modulation method. Instead of the Gaussian Minimum Shift Keying (GMSK) method used in the GSM standard, the 8 PSK method is used as can be seen in table 2.2. This enables a data transmission rate of up to 59.2 kbit/s per time slot for EDGE. Up to 473.6 kbit/s can be achieved by using eight time slots. In comparison, a maximum of 171.2 kbit/s is possible with the GPRS data service. GMSK transmits 1 bit per symbol. The symbol rate is 270.833 baud, corresponding to 270.833 bit/s. To increase the transmission speed (bps = bits per second) EDGE uses an 8-PSK (8-Phase Shift Keying) phase modulation in addition to the frequency modulation GMSK. With 8-PSK modulation, 3 bits per symbol are transmitted. Despite three times the data rate, the symbol rate is the same as for GMSK, and the pulse shape is selected so that a GMSK and an 8-PSK frequency spectrum do not differ [19].

MCS	**Data transmission rate (kbit/s/Slot)**	**Modulation**
MCS-1	8,8 kbit/s	GMSK
MCS-2	11,2 kbit/s	GMSK
MCS-3	14,8 kbit/s	GMSK
MCS-4	17,6 kbit/s	GMSK
MCS-5	22,4 kbit/s	8-PSK
MCS-6	29,6 kbit/s	8-PSK
MCS-7	44,8 kbit/s	8-PSK
MCS-8	54,4 kbit/s	8-PSK
MCS-9	59,2 kbit/s	8-PSK

Table 2.2: Overview of GPRS coding schemas (CS) and EDGE modulation and coding schemas (MCS).

EDGE can therefore transmit up to three times as many bits as GPRS in a proper channel per time unit. In a less good channel, there are frequent bit errors. The GMSK is a less fault-sensitive modulation. In the EPGRS service, which is a further development of the GPRS packet service, it is possible for the network to switch between GMSK and 8PSK depending on the quality of the channel. The optimized adaptation of modulation and coding to the channel significantly increases the data rate. EGPRS also offers all the advantages of the GPRS packet data service [20]. This means that a transmission channel is only reserved if data is sent or received. This has the advantage that several users can share the same channel.

2.3.3 UMTS and HSPA

The Universal Mobile Telecommunications System (UMTS) is a third-generation (3G) mobile communications standard that enables significantly higher data transmission rates (up to 42 Mbit/s with HSPA+, otherwise max. 384 kbit/s) than with the second-generation (2G) mobile communications standard, the GSM standard (up to 220 kbit/s with EDGE, otherwise max. 55 kbit/s with GPRS). UMTS combines the properties of a circuit-switched voice network with those of a packet-switched data network and offers a multitude of new possibilities compared to earlier technologies. UMTS technology is based on packet-oriented switching and the Internet protocol to create the conditions for mobile communication services [21]: Video telephony, Internet access, Online-shopping, Information services, Navigation. The UMTS specification provides two different modulation methods which also differ in the frequency ranges used. A distinction is made between the Frequency Division Duplex (FDD) modulation method and the Time Division Duplex (TDD) modulation method.

Frequency Division Duplex

In Frequency Division Duplex (FDD) mode, the mobile and base stations transmit in two different frequency ranges. The mobile device broadcasts in the uplink channel and the base station in the downlink channel. The two frequency ranges each have a width of 5 MHz. The individual transmission channels are realized by pure CDMA. This is a code division multiplex method that enables the simultaneous transmission of different user data streams on a common frequency range. The procedure FDD is intended for large-area radio network coverage [22].

Time Division Duplex

In TDD (Time Division Duplex) mode, the mobile and base stations transmit in the same frequency band, but at different times. A frequency carrier is divided into 15 timeslots with a total transmission time of 10 ms. Each timeslot is divided into several radio channels by CDMA. The procedure is technically more complex, since timing

Duplex Modulation	Access method	Frequency ranges Uplink	Frequency ranges Downlink
FDD	Frequency Division Multiplex	1920 ... 1980 GHz	2110 ... 2170 GHz
TDD	Time Division Multiplex	1900 ... 1920 GHz	2010 ... 2025 GHz

Table 2.3: Comparison of the frequency ranges of FDD and TDD

problems can occur if the transmitter moves or is far away from the base station. The used frequency ranges of the two different modes are listed in table 2.3[22].

UMTS (Universal Mobile Telecommunications System) has greatly increased the transmission speed. Transfer rates of up to 384 kbit/s can be achieved. This enables video telephony and other data-intensive content to be used for the first time [23].

HSPA

The UMTS radio network has been further developed over the years and offers today transfer rates in the megabit range that go far beyond the original design. These extensions are often referred to as High Speed Packet Access (HSPA).

High Speed Packet Access (HSPA) is an extension of the UMTS mobile communications standard that enables higher data transmission rates. It is divided into High Speed Donwlink Packet Acces (HSDPA) for increasing the data transmission rate of the downlink and High Speed Uplink Packet Access (HSUPA) for the uplink.

HSDPA is a data transmission method of the UMTS mobile radio standard defined by the 3rd Generation Partnership Project [23]. The method enables DSL-like data transmission rates in mobile networks. The maximum data rate is limited by the category of the receiver. Typical data rates are 3.6 Mbit/s (category 6) and 7.2 Mbit/s (category 8).The user data is transmitted in intervals of three UMTS time slots on the so-called High Speed Downlink Shared Channel. One interval has a length of 2 ms. Up to 15 channels can be assigned simultaneously. In return, a terminal device transmits information about the channel quality every 2 ms. Based on the channel quality of different end-devices and considering other parameters like buffer level and priorities the UMTS base station decides which end-devices are to be served with which number

Coding rate / Spreading factor	1 Code	2 Codes	4 Codes	6 Codes
2 / 4	0.64 MBit/s	1.28 MBit/s	2.56 MBit/s	3.84 MBit/s
3 / 4	0.72 MBit/s	1.44 MBit/s	2.88 MBit/s	4.32 MBit/s
4 / 4	0.96 MBit/s	1.92 MBit/s	3.84 MBit/s	5.76 MBit/s

Table 2.4: HSUPA Levels

of channels at the same time.

HSUPA is part of HSPA and was originally developed by Ericsson. It is a protocol add-on for UMTS to increase the transmission speed. HSDPA techniques can not be used for the uplink. This is due to the lower transmission power of the terminals compared to the base stations. This means that the signals from the end devices arrive at the base stations with a much lower quality than in the opposite direction. The low transmission power is due, among other things, to battery operation and the low antenna power of the terminals. Due to the lower transmission power, the modulation and encoding methods of HSDPA cannot simply be used. For the connection from the terminal device to the base station, more robust one-bit modulation methods are required, which are less error-prone than the higher-quality modulation methods of HSDPA.

In order to achieve a higher transmission rate, the data must be encoded with more codes, at the expense of interference immunity. If the radio connection becomes worse, then the coding rate must be decreased again. The data rate is determined by the spreading factor as can be seen in table 2.4. The greater the spread, the smaller the transmission channel and the more channels can be accommodated in the broadband transmission channel. HSUPA increases the maximum uplink speed in UMTS networks from 384 kBit/s to up to 5.76 MBit/s with optimal utilization of the spread factor and the associated coding rate as shown in table 2.4[12].

2.3.4 LTE

Long Term Evolution (LTE, 3.9G) is a term for the third generation mobile radio standard. An extension is called LTE-Advanced or 4G and is backwards compatible to LTE. For marketing reasons, LTE is already advertised as 4G and LTE-Advanced as 4G+, but from a technical point of view this is not correct. The basic scheme of the Universal Mobile Telecommunications System (UMTS, 3G) is retained for LTE (3.9G). This makes it possible to quickly and cost-effectively upgrade UMTS technology infrastructures to LTE Advanced (4G), as an example. The targeted user throughput is 100Mbps for downlink and 50Mbps for uplink, significantly higher than the existing 3G networks, with less than 5ms user-plane latency. In addition to a higher bit rate, lower latency and many other services, energy saving is also an important issue. LTE uses Orthogonal Frequency Division Multiplex (OFDM) technology, which suffers from poor energy efficiency. To save energy, LTE Uplink uses a special implementation of OFDM called SC-FDMA for Uplink, with improved energy efficiency. Discontinuous reception (DRX) was used in existing wireless mobile networks to reduce EU energy consumption [24].

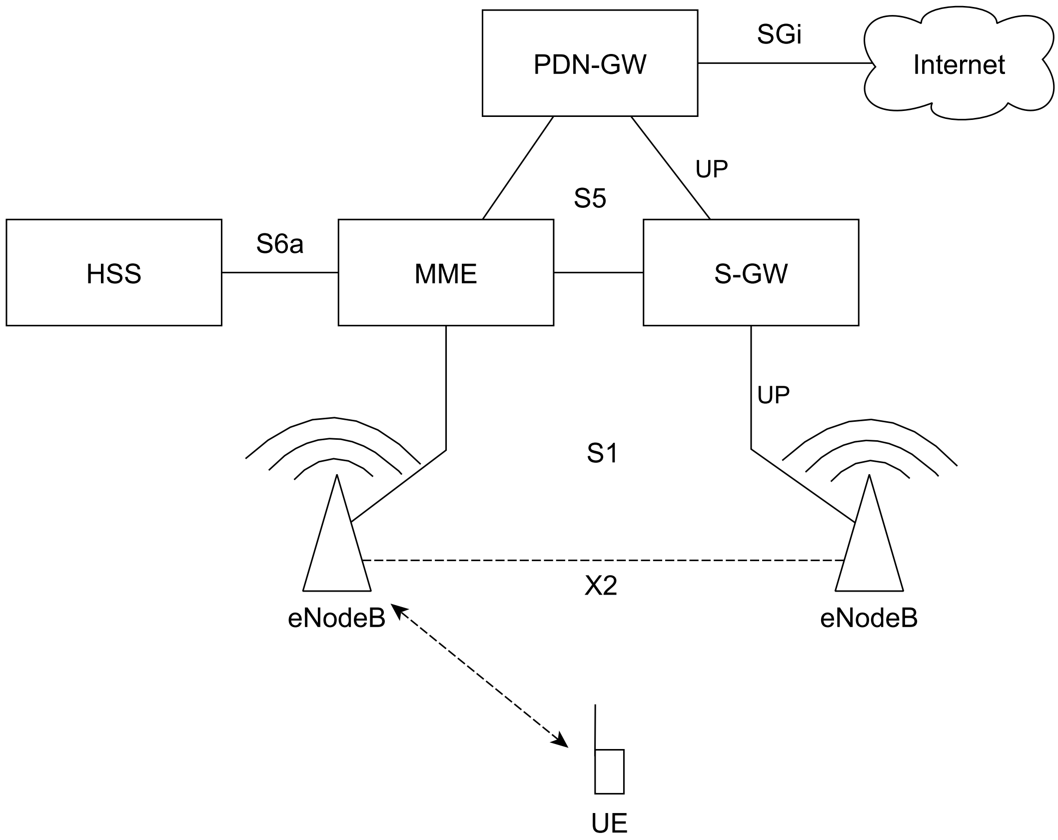

Figure 2.3: LTE overview

The architecture of the LTE network is very similar to GSM and UMTS. In the LTE network, the network is also separated into a radio network and a core network. The number of the logical components has been reduced in order to increase the efficiency, to lower costs and minimize latencies. Figure 2.3 gives an overview of the components of an LTE network and how they are interconnected [25].

User equipment (UE)

As in UMTS, terminal devices are defined as user equipment (UE) within the LTE specification. In 3GPP Release 8, the mobile device categories (UE Categories) were 1-5 in 3GPP TS 36.306 and define the maximum data transmission rate at which a terminal device can send and receive. Later 3GPP releases have included additional UE classes for devices that support carrier aggregation. Carrier Aggregation (CA) is a technique used in mobile communications to increase the data rate per user. Several individual carriers, i.e. frequency blocks, are assigned to one user. The maximum data rate per user is increased by the number of frequency blocks. The overall data rate per cell is also increased by improved resource utilization. This leads to a combination of several ranges to a virtual frequency band. For example, the aggregation of 20 MHz at 1.8 GHz and 20 MHz at 2.6 GHz results in a virtual useful band with a width of 40 MHz. LTE Advanced makes it possible for the first time to combine up to 5 component carriers into one aggregate with a maximum of 100 MHz. Each of the component carriers can be 1.4, 3, 5, 10, 15 or 20 MHz wide [12][26].

A UE category makes only one statement about the maximum supported data rate of a mobile device in downlink and uplink direction. The combination of functionalities that can achieve this speed (number of aggregated carriers, number of MIMO channels, use of 256-QAM) is not part of the definition. For all listed terminal device classes, support of at least 2x2 MIMO transmission obligatory. With this transmission technique, two data streams can be transmitted on the same channel with two antennas in the base station to two antennas in the terminal device. The number of sending and receiving antennas determines how many data streams can be sent in parallel. Most LTE terminals today have two receiving antennas. High-end devices meanwhile support four receiving antennas. Designations such as 2x2 MIMO or 4x4 MIMO indicate the number of antennas on the transmitter and receiver sides. Today however, terminal devices usually have additional antennas, since LTE is transmitted in numerous frequency bands and other radio technologies such as WLAN and Bluetooth are used in parallel.

While UMTS was initially only specified for one frequency band, LTE started in a

Category (DL/UL)	4	6	9	16/5	18/13
Max download speed (Mbit/s	150	300	450	1000	1200
Number of carriers in the downlink	1	2	3	4	4
Number of MIMO downlink streams	2	2	4	4	4
Max modulation in downlink	64-QAM	64-QAM	64-QAM	256-QAM	256-QAM
Max upload speed (Mbit/s)	50	50	50	75	150
Number of carriers in the uplink	1	1	1	1	2
Max modulation in uplink	16-QAM	16-QAM	16-QAM	64-QAM	64-QAM

Table 2.5: Selection of LTE device categories [12]

large number of different frequency bands to use resources worldwide and locally in the best possible way. Figure 2.5 gives an overview in which frequency bands LTE is mainly used in Europe today. The frequency bands that can be used with LTE are described in detail in 3GPP TS 36.101. Currently, most network operators in Europe use a combination of bands 3, 7 and 20. In the 800 MHz range in most countries, only channel widths of 10 MHz and thus data rates of around 50 Mbit/s are possible under optimum conditions in band 20. In band 3 in the 1800 MHz range and in band 7 in the 2600 MHz range, channel widths of up to 20 MHz are possible and are also used in practice[12].

eNodeB and S1 and X2 Interfaces

The most complex module in an LTE network is the base station, which is hereafter referred to as eNodeB. The name derives from NodeB, the name of a base station in UMTS. The prefix "e" stands for "evolved". An eNodeB consists of the following components:

- The antennas, the most visible part of a mobile network
- A radio module that modulates a data stream to be transmitted, demodulates received data streams, and amplifies the signals
- A digital module which takes over the actual signal processing and also acts as a digital interface to the core network serves

Transmissions between a terminal device and the base station are routed via so-called Radio Access Bearers (RAB). The RAB is assigned to a terminal during connection setup and includes the Signaling Radio Bearer (SRB) to transmit messages for Session Management, Mobility Management and Radio Resource Control (RRC). In UMTS, the base station was originally little more than an intelligent modem, whereas LTE base stations are autonomous units. Thus, an eNodeB is not only responsible for the radio interface (Air Interface), but also for:

- The user management and the allocation of resources on the air interface to several simultaneous participants
- Ensuring Quality of Service (QoS) attributes for individual connections such as maximum delay time and provision of minimum bandwidth depending on the user profile
- Mobility management
- The interference management, which means the reduction of the influence of the own broadcasting activity on the transmissions of the neighbouring stations

The theoretical top speed depends on the bandwidth of the channel. LTE is very flexible here and defines a number of different channel bandwidths from 1.25 to 20 MHz. In a 20 MHz configuration and 2x2 MIMO, data rates of up to 150 Mbit/s can be achieved. The speeds that can be attained in practice, depend on many factors, such as the distance of a terminal from the eNodeB, the power used, interference from neighbouring stations, etc. For this reason, the achievable rates are usually much lower in practice [26].

Mobility Management Entity (MME)

The user management is still a centralized task in the core network despite the autonomy of the base stations for decisions. This is necessary because the mobile radio network must have at least one fixed router which is used to transmit IP packets for a connection between the terminal device and the Internet. In addition, a user database is still needed to store authentication data and user profiles in a central location so that a terminal can log on to any point on the network and also log on to foreign networks. MME is responsible for the following tasks [12]:

- **Authentication:** When a subscriber logs in for the first time, a connection to the LTE network is automatically established before data can be transferred. The eNodeB then communicates with the MME to identify the device. The MME requests authentication information from the Home Subscriber Server (HSS), which is described in more detail below. After positive authentication, encryption is then activated on the air interface.
- **NAS Mobility Management:** If a terminal is inactive for a longer period of time (usually between 10 and 30 s), the connection via the air interface and the user's logical tunnel in the radio network is deactivated. The terminal then decides on cell change itself and there is no signalling with the network as long as the new cells are in the same tracking area. On the one hand, this saves energy in the end device and, on the other hand, reduces the signaling load in the network.
- **Handover support:** If no X2 interface is available between two eNodeBs the MME helps to coordinate the handover between the cells.
- **Structure of Bearers:** Since MME is only responsible for signalling, but not for the actual transfer of user data, one of its other tasks is communication with other network nodes to create an IP tunnel between an eNodeB and a gateway to an external network, usually the Internet.

Serving Gateway (S-GW)

The Serving Gateway (SGW) is responsible for forwarding user data in IP tunnels between the eNodeBs and the PDN gateway that connects the LTE core network to the Internet. On the radio network side, the SGW terminates the S1-UP (User Plane) GTP tunnel and on the core network side, the S5-UP GTP tunnel to the Internet gateway. The S1 and S5 tunnels for one user are independent of each other and can be changed dynamically if required during a handover. The GPRS Tunneling Protocol (GTP) belongs to a group of IP-based tunneling protocols used in GSM, UMTS and LTE mobile networks to transport GPRS packets within the network infrastructure. GTP can be divided into different separate protocols [27]:

- GTP-C for the transport of control information, e.g. for setting up and dismantling tunnels
- GTP-U for transporting the user data
- GTP for transporting data for billing charges

Home Subscriber Server (HSS)

The Home Subscriber Server is the central subscriber database in LTE mobile networks. The function of the HSS corresponds to that of the Home Location Register (HLR) in GSM or UMTS mobile radio networks. In LTE, the IP-based DIAMETER protocol is used instead for information exchange. This protocol is standardized in RFC 3588, and the interface name in LTE is S6a. In practice, however, the HLR and HSS are typically identical to allow a smooth transition between the different radio networks [27].

Packet Data Network Gateway (PDN-GW)

The Packet Data Network Gateway terminates the S5 interface. In practice, this component forms the transition to the Internet. The PDN Gateway connects the subscriber device to 2nd and 3rd generation mobile networks (2G and 3G), external packet switching networks or the Internet and forms the end point for data traffic to and from the terminal device. A subscriber device can be simultaneously connected to several PDN gateways and access several external networks. A further task of the PDN gateway is the assignment of IP addresses to terminals. During the connection establishment after switching-on, the eNodeB first contacts the MME for authentication of the terminal.

After successful authentication, the MME then requests an IP address for the terminal device from the PDN gateway. When the PDN gateway allows access to the network, it returns an IP address to the MME, which then forwards it to the terminal. Part of this procedure is also the structure of the S1 and S5 user data tunnels [12].

2.3.5 5G

5G is the 5th generation of mobile networks, a significant evolution of today's 4G LTE networks. It is being designed to meet the very large growth in data and connectivity of today's society and tomorrow's innovations. The previous four generations of cellular technology have each been a major paradigm shift that has broken backward compatibility. Indeed, 5G will need to be a paradigm shift that includes very high carrier frequencies with massive bandwidths, extreme base station and device densities, and unprecedented numbers of antennas. However, unlike the previous four generations, it will also be highly integrative: tying any new 5G air interface and spectrum together with LTE and WiFi to provide universal high-rate coverage and a seamless user experience. To support this, the core network will also have to reach unprecedented levels of flexibility and intelligence, spectrum regulation will need to be rethought and improved, and energy and cost efficiencies will become even more critical considerations. In addition to delivering faster connections and greater capacity, a great advantage of 5G is the fast response time referred to as latency. 5G will enable instantaneous connectivity to billions of devices, the Internet of Things (IoT) and a truly connected world.

Engineering Requirements

In order to clearly understand and to meet the engineering challenges facing 5G, it is necessary to first identify the key requirements for a 5G system [28].

Data Rate

The need to support the mobile data traffic explosion is the main driver behind 5G. Data rate can be measured in several different ways, and there will be a 5G goal target for each such metric:

- Aggregate data rate or area capacity refers to the total amount of data the network can serve, characterized in bits/s per unit area. The general consensus is that this quantity will need to increase by roughly 1000x from 4G to 5G.

- Edge rate in 5G technology is expected to range from 100 Mbps to as much as 1 Gbit/s. This requires about a 100x advance compared to current 4G.
- Peak rate requirements for 5G are about 100x current 4G technology.

Latency

3G networks had a typical response time of 100 milliseconds, 4G networks' typical response is around 30 milliseconds and 5G will be able to support o roundtrip latency of as low as 1 millisecond. This virtually leads to opening up a new world of connected applications.

Key Capabilities

The key performance indicators for 5G include the user experienced data rate, connection density, end-to-end delay, traffic volume density, mobility and the peak date rate. 5G systems must dramatically outperform previous generation systems, therefore, 5G technology should support [29]:

- User experienced data rate: 0.1–1 Gbit/s
- Connection density: one million connections per square kilometer
- End-to-end latency: millisecond level
- Traffic volume density: tens of Gbit/s per square kilometer
- Mobility higher than 500 km per hour

Among these requirements, the user experienced data rate, connection density and end-to-end latency are the three most important ones. Compared with 4G, 5G should have 3–5 times improvement on spectrum efficiency and more than 100 times improvement on energy and cost efficiency. The performance and efficiency requirements define the key capabilities of 5G, which are illustrated in Figure 2.4, in comparison to the 4G capabilities.

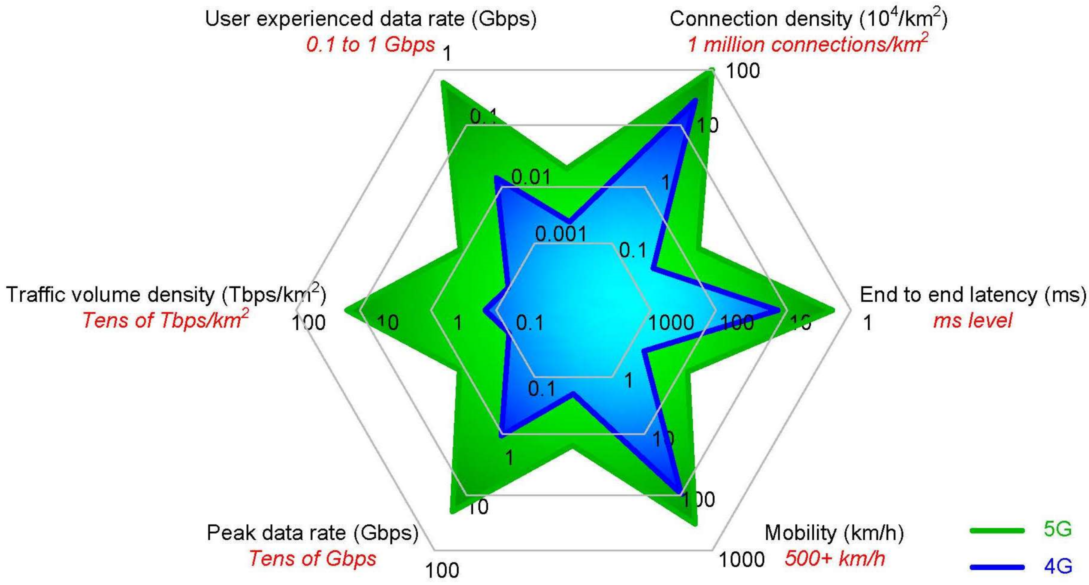

Figure 2.4: Key capabilites [29]

2.4 Tools

This section provides an overview of the tools which are used to simulate the cellular network in combination with simulated vehicular traffic. Figure 2.5 depicts Omnet++ which is used for network simulation with SimuLTE. Furthermore Veins and TraCI create an interface to the Traffic Road simulator SUMO. In this context, their characteristics and their general interaction with each other is described in the following part.

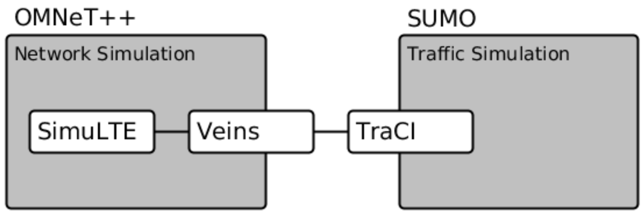

Figure 2.5: Overview of the tools and their interrogations

2.4.1 OMNeT++

OMNeT++ (Objective Modular Network Testbed in C++) [30] is a discrete event simulation framework based on C++ for modeling communication networks, multiprocessors and other distributed or parallel systems [31]. Therefore, it is used for discrete event systems, which are systems where state changes (events) occur at discrete places in time and events themselves do not need time to happen. It is assumed that nothing of significance happens between two consecutive events. In contrast, continuous systems are systems where state changes are continuous [32]. This Eclipse-based development environment is open-source and intended for use in academic, educational and research-oriented projects, as in case of this thesis.

Due to its generic architecture, OMNeT++ can be used for any application that can be modeled with the approach of discrete events and message exchange such as:

- modeling wired and wireless communication networks
- protocol modeling
- modeling of multiprocessors and other distributed hardware systems [33]

OMNeT++ itself is not a simulator but a simulation framework which provides the basic machinery and tools to create simulations. In order to achieve large scalability through hierarchical and reusable components, OMNeT++ relies on a model structure that consists of modules communicating through message exchange. There are two types of modules to be distinguished, the *simple modules* and the *compound modules*. A *simple module* is an active module which is written in C++ via the simulation class library. Whereas a *compound module* consists of multiple *simple modules* or other *compound modules*, while the whole model, called *network* in OMNeT++, is a *compound module* as well. Messages can be sent either directly to other modules or via the connections between two modules.

This structure is displayed in Figure 2.6, where the gray boxes represent *simple modules* and the white ones represent *compound modules*. The connections are represented by the arrows and connect the different gates which are shown as small black boxes.

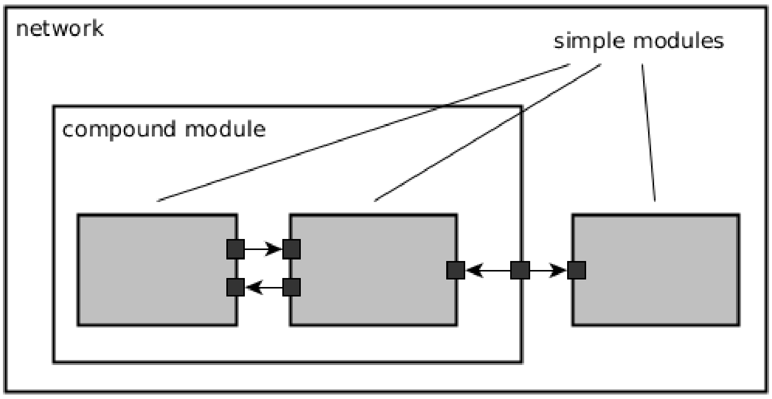

Figure 2.6: OMNeT++ Model Structure.

While the behavior of the models is captured in C++ files, while its topology defining parameters are stored in NED files. These NED files describe the structure of the model with their modules and their interconnections using their own topology description language NED. It was designed to scale well and to make the inclusion of external models easy and straightforward.

The general design of OMNeT++ separates the above-mentioned models with the simulation experiment itself. The parameters for the simulation are stored in INI files, which provide a way to specify and combine parameters for different simulation configurations [31].

OMNeT++ Frameworks

The reusability of OMNeT++-based frameworks is another reason for its usage in this thesis. These frameworks are developed independently of OMNeT++ itself [31]. The widely used model library INET provides standard protocols and models for wired, wireless and mobile networks. These are e.g. TCP, IPv4, Ieee802.11b/g and several other protocols. Many frameworks are based on INET [34]. These include Veins for vehicular networks and SimuLTE for LTE based applications. Both frameworks are used for this thesis and will be described in more detail in the next sections.

2.4.2 SimuLTE

As mentioned before, SimuLTE is a OMNeT++ framework based on INET. It provides the data plane of LTE and LTE-Advanced networks. The simulations base on Frequency Division Duplexing (FDD) mode, which means that there are two different frequencies for up- and downlink transmissions [35]. Heterogeneous eNodeBs as well as UEs are modeled as compound modules, which will be discussed in more detail in the following part.

Nodes

LTE and LTE-A networks consist of eNodeBs and UEs. These two in combination with the Binder represent the main nodes of SimuLTE. Their general structure can be seen in Figure 2.7. In order to create networks, eNodeBs and UEs can be connected with each other and with other nodes (e.g. routers, applications, etc.). While these nodes are only visible among each other, the Binder module is visible from all nodes of the system and contains information about them [36]. The Binder stores references to nodes and monitors which resources, i.e. Resource Blocks (RB), are used. It can be considered as a monitoring unit.

Both in UEs and in eNodeBs there is an UDP and a TCP module from the INET framework containing the respective transport layer protocols. These modules connect the LTE stack to TCP/UDP applications. As shown in Figure 2.7 multiple applications are possible which are pictured as vectors of N modules. This must also be the case since each TCP/UDP app represents the end of the respective connection.The IP module, also taken from the INET framework, acts as connection between the Network Interface Card (NIC) and the applications using TCP or UDP. In case of the eNodeB it also connects to the Internet via Point-To-Point-Protocol (PPP) or to other eNodeBs via the

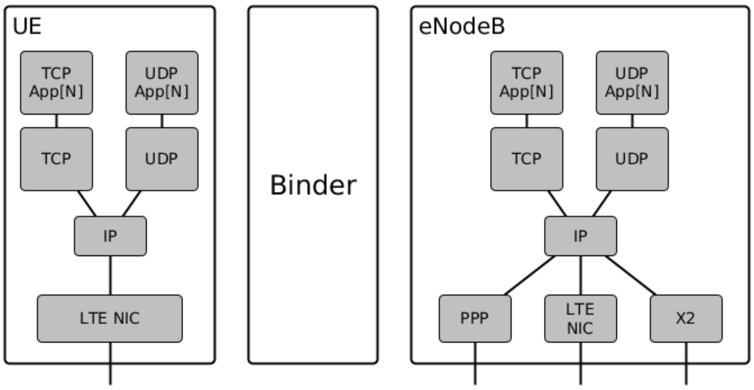

Figure 2.7: UE and eNodeB module structure.

X2 interface[37]. The X2 protocols provide the base for functions such as Coordinated Multi Point or Network Assisted Handover[38].

LTE NIC module

The LTE NIC implements the whole LTE protocol stack with one submodule per layer, as it is shown in Figure 2.8. It therefore consists of four submodules: Packet Data Convergence Protocol (PDCP)/ Radio Resource Control (RRC), Radio Link Control (RLC), MAC and PHY. Due to the inheritance paradigm of OMNeT++ SimuLTE provides common operations for both UEs and eNodeBs, which are extended with their respective functionalities in the submodules[37].

The PDCP/RRC module is the connection between LTE NIC and the IP modules. In the downstream this module receives data from the IP module and accordingly in the upstream mode from the RLC module. The PDCP part is responsible for the header compression/ decompression. On the other hand the RRC part performs tasks like assigning a Logical Connection IDentifier (LCID) for each connection as an unique identification and binding of the IP Address with its module id for the Binder module.

The RLC module does multiplexing and demultiplexing of MAC service data units. For the transmission of data the three RLC modes are implemented: Transparent Mode (TM), Unacknowledged Mode (UM) and Acknowledged Mode (AM).

The MAC module is the one where most of the intelligence resides. As an overview

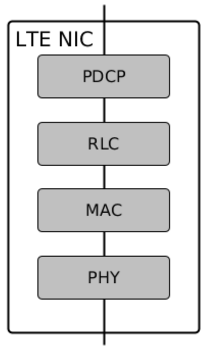

Figure 2.8: LTE NIC module architecture

the most important functionalities are the following:

- packet buffering from RLC and PHY
- Hybrid Automatic Repeat Request (H-ARQ) functionalities
- scheduling for eNodeBs
- managing channel feedback

The last layer of the LTE-NIC module to be described is the PHY module, which implements functions according to the physical layer. It supports heterogeneous nets consisting of Makro-, Mikro or Pico-eNodeBs. Besides it is responsible for the channel feedback management as well as data transmission and reception. The air channel emulation is dependent on the associated Chanel Model. It calculates the Signal to Interference plus Noise Ratio (SINR) and checks for corrupt packages as base as a basis for further proceeding with air transmissions between LTE NICs. SimuLTE therefor implemented the so called *Realistic Channel Model* to generate, as the name implies, a realistic SINR value. The calculation of the SINR in this model considers cell interference, transmission power, path loss, fast fading and shadowing among others [36].

2.4.3 SUMO

For the traffic road simulation SUMO is used, which is the abbreviation for Simulation of Urban MObility. SUMO is an open source, space-continuous and time-discrete traffic simulation package[39]. It is a microscopic simulation tool, which means that each vehicle and its dynamics are modeled individually. This is an important reason for using SUMO, as it uses this precise model, which allows the routes of each vehicle to be determined individually [40]. SUMO provides not only of the simulation itself, it also consists of several applications to prepare the traffic simulation. There are two main steps to create a traffic scenario: the network generation with its traffic infrastructure (e.g. traffic lights) and the traffic generation consisting of vehicles and their routes.

Road Network

For the road network generation SUMO provides two applications: NETEDIT and NETCONVERT. NETEDIT is a graphical network editor which allows manual creating, editing and analyzing network files. The other application for the network generation is NETCONVERT. It is a command-line tool which converts road networks from existing formats such as OpenStreetMap (OSM)[41] or OpenDRIVE[42] to its own format.

Both applications lead to SUMO road networks which are encoded as XML files[39] that consist of the following network elements:

- **Basic network elements**: junctions, edges (roads) and lanes with information such as maximum speed. Besides, there are also Connections, which contain information about how the edges of junctions are connected.
- **Advanced network elements**: These elements have a direct influence on traffic behavior, e.g. traffic lights .
- **Additional infrastructure**: Elements which do not belong to the network itself, but may be used to influence the simulation. E.g. bus stops or detectors for saving information about passing vehicles.
- **Poligons and points of interest (POI)**: These elements are visual representations of objects which do not influence the simulation such as buildings[43].

Traffic

For the creation of traffic, the previously created road network is required as the base. It is possible to assign an individual route to one or more vehicles, where a route consists of a series of edges. SUMO therefor provides applications for the creation and adaption of traffic consisting of vehicles and their routes among other things. Vehicles are given a departure time and an associated route. SUMO also allows an route adaption during the simulation, which enables an interactive recalculation of the routes. The intern function then chooses the fastest route to its destination edge according to the edge weights of the route. The edges are weighted by their travel speeds and the resulting travel times[40]. As SUMOs routing is based on the shortest path problem, it offers four algorithms for selection: Dijkstra (default), A*, Contraction Hierarchies (CH) and CHWrapper [44]. These algorithms are described in detail in Section 2.5.

TraCI

The Traffic Control Interface (TraCI) is an API that offers the possibility to retrieve simulation objects and to modify them during simulation runtime. Among other things, it enables interaction with objects relevant to this thesis such as routes, edges, lanes and vehicles. This allows actions such as blocking lanes, changing a vehicles destination or adding routes. TraCI supports both in-line interaction to couple two or more SUMO simulations, and on-line interaction to interact with external applications. It uses a client/server architecture which allows interaction with any application supporting TCP sockets. Therefor SUMO (server) needs to be started and specified with a port number to listen to. Besides interfaces for Python or Java, there is also a middle-ware solution for the coupling of OMNeT++ and SUMO which is called VEINS [39]. This is described in more detail in the following part.

2.4.4 Veins

With OMNeT++ for the network simulation on the one side and SUMO for the traffic simulation on the other side, a tool is needed to facilitate the interaction with each other. Veins, which stands for *Vehicles in Network Simulation*, enables exactly this. It is an OMNeT++ based framework for Car2X communication that provides simulation models for vehicular networking and interacts bidirectionally via TraCI with SUMO[45].

2.5 Route Determination

The problem of route determination is ubiquitous. The everyday task to get from one location to another, needs a route to get there. Since there are restrictions due to the structure of the road network, the possibility of the direct linear line as a route is excluded. The goal is to find the shortest, quickest or cheapest route. In graph theory the shortest-path problem deals with exactly this problem. In order to use the algorithms for solving this problem, the road network must first be converted into a directed graph *G = (V,E)* with a cost function *c* where

- V is a set of nodes
- E is a set of directed edges with $E \subseteq V \times V$
- c is the cost function that maps the edges to arbitrary real-number costs with $c : E \rightarrow \mathbb{R}$ [46]

2.5.1 Road Network to Directed Graph

For the purpose of generating a directed graph from the road network, the intersections or end of roads become nodes and the streets become edges (similar to SUMO). An example is depicted in Figure 2.9 (destinations between intersections are not considered here for the sake of simplicity). The nodes are labeled v_1 to v_{25} and connected via edges. These edges correspond to bi-directional roads with arrows at both ends, which can be driven on from both sides. A connection with only one arrow corresponds to a one-way road, such as the edge (v_3, v_{10}).

The numbers above the edges correspond to the result of the cost function. In this case it has only one value per edge for clarity, even with bidirectional edges. The cost function can, according to its definition, depend on information like distance, time, flow, street availability etc. In the simplest case, as in this example, this corresponds to the distance between the two nodes[47].

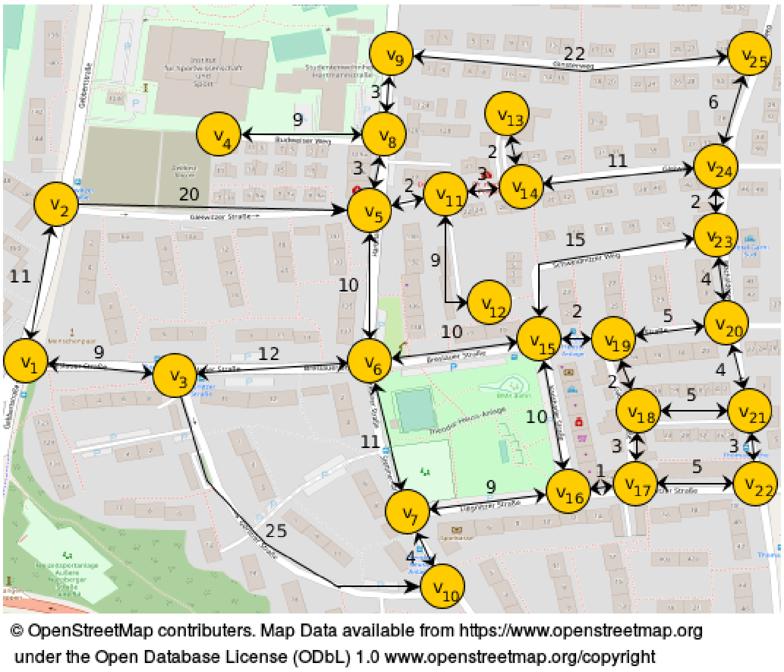

Figure 2.9: Resulting graph model laid over a road network in Erlangen by OpenStreetMap [41].

2.5.2 Shortest Paths

The cost function can be adjusted according to requirements, such as the shortest or fastest route. The cost of a path p is the sum of its constituent edges:

$$p = \langle e_1, e_2, ..., e_k \rangle \in E \times E \times ... \times E$$

with

$$c(p) = \sum_{i=1}^{k} c(e_i)$$

The empty path has cost of zero.

In the case of road networks, a distinction is made between static and dynamic networks. This means that in the case of static networks there is a fixed value for each cost of an edge. In the case of dynamic networks, on the other hand, the cost function can take into account traffic jams, for example, which can lead to a change in costs [48]

One is now searching for the best path from one node to another corresponding to the lowest cost from all possible paths, the so called shortest path. Since there are several algorithms for the shortest path problem, the four algorithms implemented in SUMO are now explained in more detail.

For the application of the algorithms we consider nonnegative edge costs, integer edge costs and acyclic graphs. The following graph in Figure 2.10 serves as a basis for the following algorithms[47].

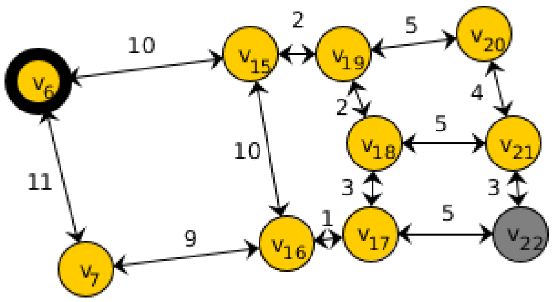

Figure 2.10: Subgraph of the graph in Figure 2.9 with start node v_6 and destination node v_{22}

Dijkstra's Algorithm

Dijkstra's Algorithm is the classical algorithm for route determination. At beginning of this procedure, every node gets the tentative cost of ∞ except v_s, which gets cost zero. The algorithm visits the nodes of the graph in the order of their cost to the start node. This means to get the lowest cost path, one has to find the minimum cost for each path from the start node v_s to each node $v_c \in V$. To do this, a set S of nodes is observed and gradually enlarged, for which the shortest path from v_s is already known. Therefore one take a look at the connected nodes and search the one with the lowest edge costs and add it to S. For every edge from the current node v_c to nodes which are not in S, one updates the tentative cost of these nodes represented by v_n with:

$$c(v_s, v_n) = min(\{c(v_s, v_n), c(v_s, v_c) + (v_c, v_n)\})$$

This algorithm continues then with the next lowest cost edge and terminates when the destination node v_d is visited. As there are two arrays indexed by each node n nodes and it can be at most m cost update operations Dijkstra's Algorithm can lead to $\mathcal{O}(m + n^2)$ [49].

A* Algorithm

The A* algorithm is very similar to the Dijkstra algorithm. It differs essentially in the informed algorithm, which uses a heuristic for the selection of the next node to be regarded. For the selection of the edge it uses the result of the evaluation function f, which depends on the cost function c and the heuristic function h. With v_s as start node, v_n as a connected possible next node and v_d as the destination node, f is defined as follows:

$$f(v_n) = c(v_s, v_n) + h(v_n, v_d)$$

The cost function c returns the cost for the path from the start node to the current node and the heuristic function h returns an estimated cost of the cheapest path to the destination node. By replacing the heuristic function with zero, the A* Algorithm would be the same as Dijkstra's Algorithm. Thats the main reason why this algorithm is at least as fast as Dijkstra's[50].

Contraction Hierarchies

The method of Contraction Hierarchies (CH) is based on the concept of node contraction. First of all the nodes of the graph have to be sorted by their 'importance', starting with v_1 as the least important and ending with v_x as the most important one [51]. A hierarchy is now created by contracting nodes in ascending order. Contracting means that a node v_b is removed from the network, if the lowest cost path $p = (v_a, v_c)$ is the same as $p' = (v_a, v_b, v_c)$. For this shortest path then a new edge (v_a, v_c) will be added. This means that after contracting the nodes, the remaining overlay graph consists of new edges which allow a much faster calculation of the shortest path [52].

3 Methodologies

In this chapter different approaches for mobile network availability dependent route determination are explained and their characteristics are described. All approaches are based on a known and unrestricted road network. Due to the fact that a stable connection is required for teleoperated driving, individual areas are blocked, based on the mobile network availability. The roads located in this areas are closed for the resulting road network, since a (stable) connection to the **remote driver** would not be possible. The resulting restricted road network then should enable to define routes by any route determination algorithm, which can then be driven on by teleoperated vehicles.

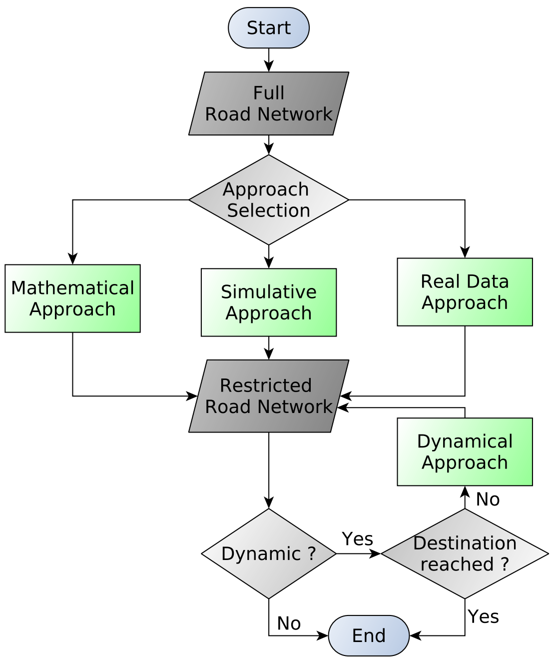

Figure 3.1: Flow chart of road network adjustment with different concepts.

The flow chart in Figure 3.1 provides an overview of the different approaches for adapting the road network. Starting with a full and unrestricted road network, the restricted road network is then created using a Mathematical, Simulative or Real Data Approach. The resulting static road network can then be used as the basis for route determination. In the case of the Dynamic Approach, the already restricted road network is updated using real-time information until the destination is reached. The updated network can then be used to adjust the route during the vehicles routing process.

In the scope of this thesis the focus will be on the Mathematical and Simulative Approach, which will be explained in detail in the following part.

3.1 Mathematical Approach

The focus of this method is to give a rough approximation to the actual availability of Cellular Network in a selected area. This mathematical approach is a theoretical solution where optimal conditions prevail. In the case of cellular networks this means that some simplifications are assumed which enable a fast evaluation of road networks. The main assumption is that within a certain radius around a base station an UE can communicate completely unrestricted via the Cellular Network. In this radius there are optimal conditions, since any kind of interference factors are neglected.

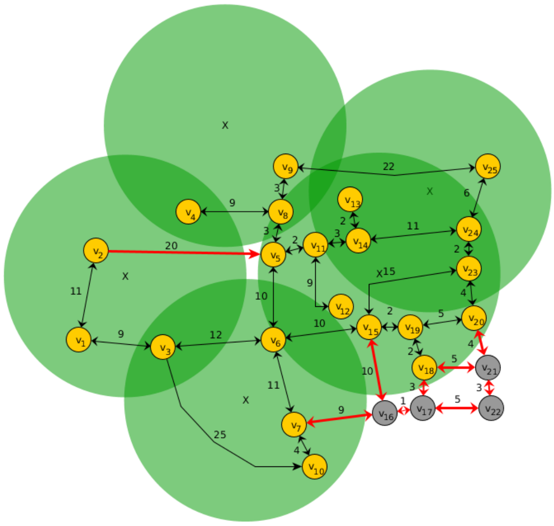

Figure 3.2: Cellular network coverage on street network example graph.

As can be seen in Figure 3.2 the range and the position of the base stations are represented by the green circles and their centers. These circles are now layered over the known road network (represented as a directed graph) according to their position. In the next step, all roads are blocked for the resulting road network that are not completely within the covered area. This means that as soon as even a minimal part of a road is outside, the entire road has to be blocked. These roads are shown in the Figure by red edges. In this procedure the resulting road network depends only on the position and the range of the base stations. The choice of a suitable radius thus plays a large role. A too large radius e.g. means less blocked roads and consequently a less restricted road network, which in reality could lead to a connection loss of the teleoperated vehicle.

An example of how a route in an unrestricted road network might look like compared to a route in a restricted one can be seen in the following figures. In both cases, the shortest route from start point v_2 to destination point v_{24} is displayed as green colored edges. Figure 3.3 shows the direct route for normal vehicles, whereas Figure 3.4 shows the route for teleoperated vehicles depending on the previously calculated blocked non-drivable roads.

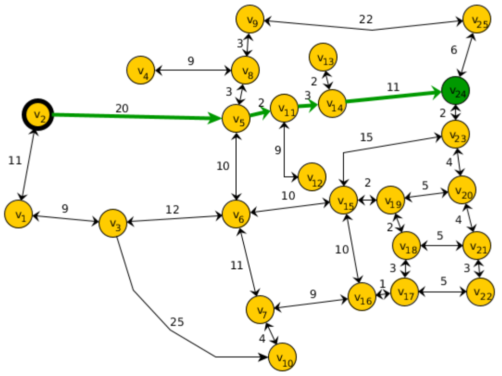

Figure 3.3: Standard route without restrictions

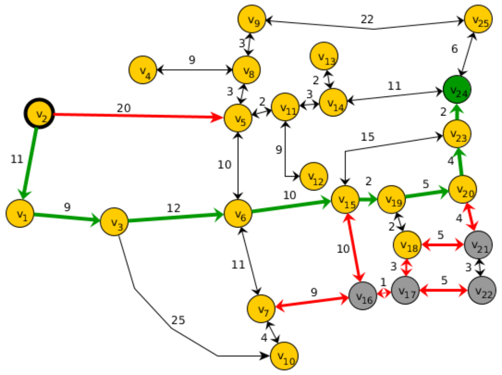

Figure 3.4: Resulting route with restriction by non-drivable roads

3.2 Simulative Approach

In this approach simulated data will be used to create a road network that can be used for teleoperated vehicles. The focus here is on a more realistic mapping of the cellular network availability on the road network under observation of several influencing factors. In contrast to the mathematical approach, interference factors can be considered by means of a simulation, such as the transmitter antenna height, free-space path loss, building attenuation, etc. The more factors are considered in the simulation of the cellular network, the more accurate the prediction of availability becomes.

The following Figure 3.5 shows the already known street network graph with the corresponding base stations. The signal propagation wit the free-space path loss is indicated by the saturation of the green color around the base stations whose positions are determined by crosses. The black rectangle represents a building with the resulting attenuation.

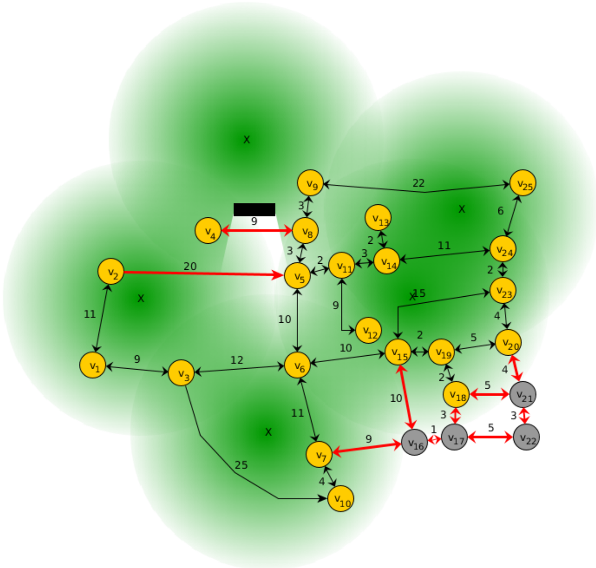

Figure 3.5: Cellular network coverage basis for restricted street network on example graph.

Based on this simulated cellular network, the next step is to let one or more measuring vehicles drive through the entire road network while constantly checking the connection to the cellular network. The availability of the cellular network is guaranteed if the error-free transmission of messages over the entire stretch of a road is ensured. Depending on results for traveled coordinates, the road is enabled or blocked in the resulting road network as in the mathematical approach.

4 Implementation

This chapter deals with the implementation of the approaches presented above for the creation of the restricted road network for teleoperated vehicles. To produce a comparable result, in both cases the road network of the Erlangen Scenario is taken from the Veins examples [45]. The focus of this thesis is on the cellular network LTE. At the time of this thesis it is the current cellular network standard with the highest data rate (as described in 2.3 Cellular Network) and the infrastructure is already in place in contrast to 5G [53]. In the following sections will be described which tools have been used and especially how they have been used to realize the approaches.

4.1 Environment

The presented approaches differ mainly in one point, how the decision is made whether a road is enabled or blocked for the restricted road network. Based on this, first the shared pre-processing for the realization of both approaches is described and then the specific tools and implementations of the mathematical and simulative approaches. This includes the creation of the input road network, the data about the LTE base stations and for the following evaluation a tool for route determination and a vehicular traffic simulation.

The whole project was build on the Linux distribution Ubuntu 16.04 LTS. To generate a comparable result between the two approaches, SUMO serves as a common basis. SUMO, used here in version 0.30.0, supports traffic simulation and provides tools for the creation of the simulation itself. In the focus of this thesis are the following tools:

- **NETCONVERT:** A helper tool provided by SUMO to convert existing road networks to files in the SUMO network format.
- **SUMOlib:** A Python library for working with SUMO networks, simulation output and other simulation artifacts.

- **DUAROUTER:** A SUMO helper tool as well which creates vehicle routes based on the shortest-path computation [40].

Due to the fact that SUMO with SUMOlib provides a Python library with many useful modules, Python 2.7.12 is used accordingly to adapt the SUMO road network files.

4.1.1 Road Network of Erlangen

As just mentioned, the input for both approaches is the SUMO road network file of Erlangen. It is based on a map which shows a part of Erlangen around the Friedrich-Alexander-University Erlangen-Nürnberg.

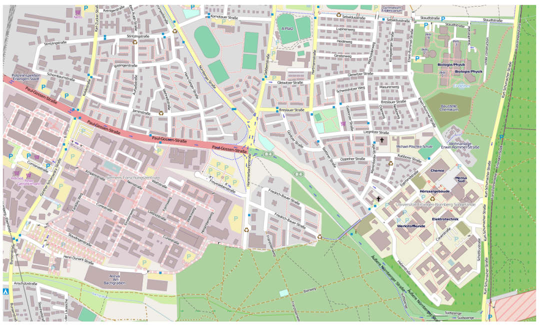

Figure 4.1: Erlangen road network from OpenStreetMap.

Figure 4.2: Erlangen road network visualized in SUMO-GUI.

To convert an existing road network into a road network compatible with SUMO, SUMO offers the command line application NETCONVERT. It enables among others the conversion from an OpenStreetNetwork map to a SUMO road network file as it was done here. In Figure 4.1 the relevant area is displayed as a map from OpenStreetMap. It shows the area to be analyzed for the cellular network availability. The SUMO road network file created by NETCONVERT and the OpenStreetNetwork data is visualized by the SUMO-GUI and can be seen in Figure 4.2. The resulting road network is the one that is used as the input base road network for both approaches. This resulting file is a representation of a directed graph in human-readable XML format. It consists of the following tags and attributes which are displayed in Table 4.1 which provides an overview of the tags used. Since some tags are not available in the Erlangen road network file like the one for the traffic light logic or edges describing crossings, these are considered in the following table.

Tag	Required Attributes	Optional Attributes
net		version, xmlns:xsi, xsi:noNamespaceSchemaLocation
location	netOffset, convBoundary, origBoundary, projParameter	
edge *(normal)*	id, from, to	priority, function
edge *(internal)*	id, function	
lane	id, index, speed, length, shape	allow, disallow
junction *(plain)*	id, x, y, incLanes, intLanes, shape	z, customShape
junction *(internal)*	id, x, y, incLanes, intLanes	z
request	index, response, foes, cont	
connection	from, to, fromLane, toLane, via, dir, state	tl, linkIndex

Table 4.1: SUMO network file elements

The net element is the root element of the whole network and encloses all other elements. Its available attributes are for XML schema validation purposes. These attributes are set by any SUMO application automatically, but can also be removed if no validation is wanted.

The next element is the location element which is a childless one. SUMO is using cartesian, metric coordinates where the leftmost node is *x=0* and the most at the bottom is at *y=0*. The process for alignment is documented within this element.

Edges are distinguished between normal and internal edges. The normal normal edges always have at least one lane as child element and correspond to a road between two junctions. Internal edges consist as well of one or more lanes and lie within an intersection. These internal edges are located between incoming and outgoing edges but are not displayed in the SUMO-GUI.

The lane element is the core element for this thesis. Besides the fact that it describes the shape by exact coordinates and defines the maximum speed of the lane, it is also possible to allow or disallow vehicles to use these lanes. Either special vehicle types or all vehicles can be allowed/disallowed.

Further elements are junctions which correspond to nodes. A distinction is made between plain junctions or internal junctions. Plain junctions describe the known street crossings and consist of request child nodes. These requests describe the behavior of the vehicles at the intersection. This allows the implementation of any right-of-way rules. The internal junctions on the other side which are responsible for defining the waiting position for vehicles within the intersection. It is typically used for left-turning vehicles. The connection element connects lanes at junctions. It describes which outgoing lanes can be reached from an incoming lane.

The resulting road network in SUMO XML format can now be imported and processed in Python using SUMOlib, which is described in more detail in the following sections.

4.1.2 LTE Cell Data

The OpenCellID database provides information about cells and their location. It is the largest open source database and the result of a community project by Unwired Labs. As the locations of most cell towers are not publicly accessible **cellTower** OpenCellID lets collect the data mainly through the community using smartphone apps for data measuring. Other data sources are for example donations from other databases or even some GSM network operators who provide the exact location of their base stations [54].

The database file containing the cells located in Germany (Mobile Country Code: 262) provided by OpenCellID is called *262.csv* and has a large amount of data, which makes it advantageous to reduce it. For this purpose the Linux command-line editor sed (**s**tream **ed**itor) was used to remove the unwanted entries of cells that are not for a LTE network [55]. The resulting file then was imported to a local MySQL database to enable working with this dataset [56].

radio	mcc	net	area	cell	lon	lat	range	changeable
LTE	262	2	48045	21109506	11.001434	49.575477	2513	1
LTE	262	3	50417	784133	11.013565	49.580612	1000	1
LTE	262	3	50417	784132	11.015167	49.585419	1000	1
...	...	...	...	...	...	...	...	...

Table 4.2: Provided Data by OpenCellID

The database itself contains several data fields, of which the relevant ones for this thesis with some example entries can be seen in Table 4.2. These include *radio*, containing the type of cellular network technology, and *mcc* with the mobile country code. These fields always contain "LTE" and "262", since there are only LTE base stations from Germany. The **net** field is for the Mobile Network Code (MNC) which is a number representing a network provider. **Area** contains the Location Area Code (LAC) and *cell* with the unique Cell-ID. The position is indicated as geographic coordinates by *lon* (longitute) and *lat* (latitiude). If **chageable** has the value "0", this means that the given coordinates correspond to the real coordinates. Otherwise, if it has the value "1", the given coordinates correspond to the average value of the coordinates that occurred during measurements of this cell. The last field to be considered is *range*. It contains information about the approximated range of the signal provided by the base station [54].

Since SUMO uses Cartesian coordinates, two more fields named x and y have been added to the local database, corresponding to the SUMO coordinates. To add these values the Python script *x-y-updateDB.py* was used. With the help of SUMOlib, the Geo-Coordinates area is first determined by the Erlangen road network and then the base stations in this area are updated with the corresponding SUMO coordinates.

OpenCellID mainly collects MCC, MNC, LAC, and CID information of a cell in combination with a GPS position. Since this data is mainly obtained by community measurements, this could lead to several localization errors:

- **Erroneous Cell IDs:** This is caused by a mix up of Cell-ID, Local Area Code and Mobile Network Code.
- **Antenna dragging:** This happens when a device does not update the Cell-ID

during a trip, even though it has changed and reports the original Cell-ID.

- **Outliers:** These are registered measurements in which the position is far away from the real position.
- **Unrealistic cell sizes:** The measurements leading to a single cell is distributed across a big area, which is clearly unrealistic **openCellId2**

Network Adaptations

Due to the fact that the available data does not provide any information about the number of sectors (cells) or the position of a eNodeB, each database cell entry is represented as a base station (eNodeB) with exactly one omni directional antenna. This enables the best possible reconstruction of the results of the measurements from the OpenCellId database. The range is also set to a common value to reduce misconduct and to get a more realistic environment. Since the most occurring value corresponds to a range of 1000m, which would lead to a fully covered area of Erlangen as shown in the left image of Figure 4.3, this value was reduced to 500m. This results in a partial net coverage as shown on the right image of Figure 4.3.

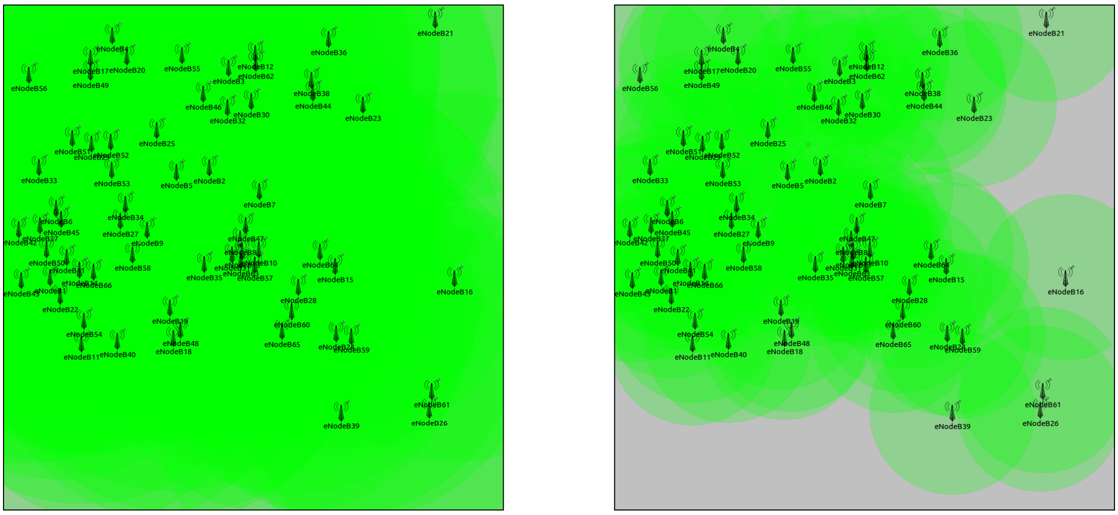

Figure 4.3: Adaption of the rang from 1000m (left) to 500m (right).

When comparing the two images, the gray area displaying the places uncovered by the cells can be seen, thus restricting the area in which a teleoperated vehicle can be driven. For the purpose of this thesis, setting the value at 1000m would correspond to a full coverage, what would cause no difference in determining the route for a teleoperated vehicle, as opposed to a normal one.

4.2 Mathematical Realization

This section shows a possible implementation of the Mathematical Approach with its resulting road network of the Erlangen area. As mentioned before, the SUMO road network file resulting from 4.1.1 and database from 4.1.2 as base station data source are used for the realization.

4.2.1 Implementation

This approach was implemented by using a Python script. Since a SUMO road network file is being modified, the previously converted Cartesian coordinates from the database are also used for the localization of the base stations. This corresponds to the values x and y of a database entry. It is mainly divided in 3 steps:

- **Step 1:** Fetch base stations from database and lanes from Erlangen SUMO road network file.
- **Step 2:** Set signal radius and check if the entire lane is within the radius.
- **Step 3:** Create a new SUMO road network file with blocked lanes for teleoperated vehicles.

Now that one has an overview of the implementation, the individual steps are explained in more detail. The first step describes the data import to Python. For the road network SUMOlib was used, which initialize data structures and create a net object containing nodes, edges, lanes and connections. After having created the object one can request the boundaries of the net object, which are used as condition for querying the respective base stations from the local mySQL database.

Step 2 checks the theoretic availability of LTE signal on every lane within the net object. All base stations located in this area are iterated for this purpose. For each base station, the edges located in the previously defined 500m radius are then retrieved from the net object. These edges contain the lanes with their shape represented as a series of coordinates. From this it can now be derived by calculating the distance between each shape coordinate and the coordinates from the base stations, whether the entire lane is within the radius of the base stations. Each lane fully covered by LTE signal gets stored in an array.

The final step is to create the resulting SUMO road network file. For this purpose the built-in Python file handler opens the road network XML file and saves the corresponding modifications in a new SUMO compatible XML file called *mathematicalAvailable.net.xml*. Each lane which is not in the previously defined available lane array is blocked for the resulting network file. This is accomplished by setting the disallow attribute to *custom1*, which blocks the lane for vehicles of same-named type. The *custom1*-type was chosen, as no teleoperated vehicle type is predefined by SUMO.

4.2.2 Resulting Road Network

Figure 4.4: Resulting route network of the mathematical approach visualized in NETEDIT.

The resulting road network based on the Mathematical Approach was opened with NETEDIT in order to be able to represent it visually, as on can see in Figure 4.4. The black colored edges are available routes, which can be used by teleoperated as well as normal vehicles. The red edges, on the other hand, are outside of the signal area and thus not accessible for teleoperated vehicles.

4.3 Simulative Realization

This chapter outlines the technical aspects needed to determine routes for teleoperated cars based on LTE network communication. For this purpose the Simulative Approach with its setup is described. The aim of the simulation itself is to get the coordinates where a car is connected to the LTE network. To be able to check this, a server constantly sends data via UDP to target cars which move through the environment, based on the information of the vehicular simulation. Whenever an error-free packet arrives, the actual coordinates of the car are stored. After the cars traveled the entire road network, the stored coordinates are placed over the road map. Based on this, it is then derived whether a road is available for the resulting road network or not.

4.3.1 Extended Environment

The entire simulation is based on an interaction of different frameworks which are embedded in one another or communicate with each other. The first challenge was to find a simulation environment which can do both, simulate cellular networking and vehicular traffic. The combination of realistic cellular network communication and realistic traffic simulation based on existing road networks. OMNeT++ is used with the SimuLTE framework allowing LTE communication, which is based on the INET framework, as already mentioned in 2.4.2. The vehicular simulation runs over SUMO, which communicates via the VEINS framework to the OMNeT++ based simulation. The whole project was build on the linux distribution Ubuntu 16.04 LTS. Due to compatibility requirements of Veins v4.6 [57], the following framework versions were used:

- OMNeT++ v5.1
- INET v3.6
- SUMO v0.30.0

4.3.2 Cellular Network Creation

This section deals with the creation of the cellular network LTE with all required components. The modular and component-based C++ simulation framework OMNeT++ is the core of this simulation setup. As described in 2.4.1 it allows to use various modules for wireless communication networks and different protocols.

First the individual submodules which were used are described and then in 4.3.3 the corresponding parameters used according to configure them the simulation.

Submodules

The submodules of the network named *ErlangeLTE* are defined in the .ned-file with the same name. It consists of several submodules which are required for the realization of the network. An overview of the used submodules and some connections is given in Figure 4.5. This shows the direct submodules of the network before the start of the simulation. These submodules can be divided into four groups according to their functionality:

- Environmental Modules
- Internet Modules
- LTE Modules
- Vehicular Modules

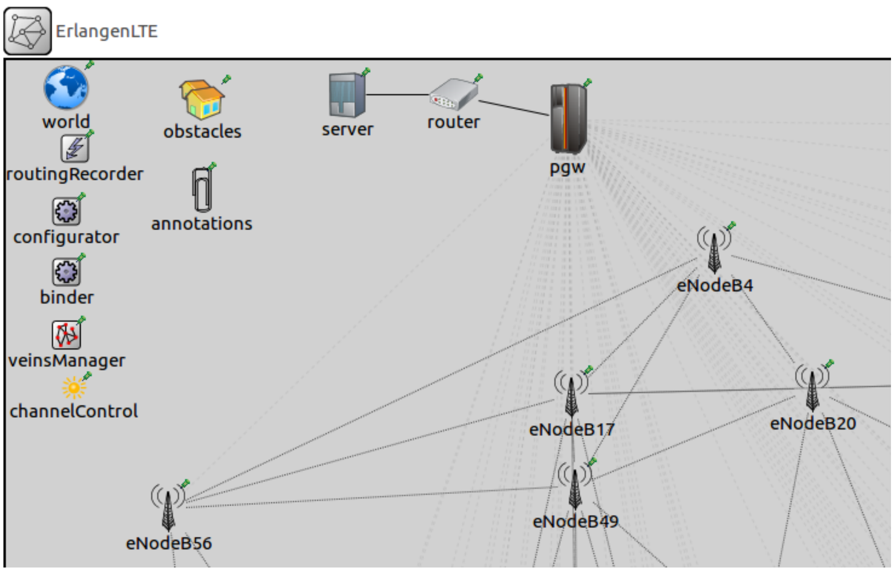

Figure 4.5: OMNeT++ network ErlangenLTE with its main submodules.

The modules belonging to the Environmental group are part of the Veins framework. The first module to describe is BaseWorldUtility (*world*), which is the basic utility module and provides the parameters of the network borders for each module. The AnnotationManager (*annotations*) gives a visual notification during the simulation for example if a handover took place. Furthermore there is the ObstacleControl module (*obstacles*). It visualizes the obstacles like buildings which do also interfere radio transmissions.

The Internet Modules are from the INET framework and provide the elements for simulating an IPv4 network. This includes the IPv4NetworkConfigurator (*configurator*) which assigns IP addresses and sets up static routing for an IPv4 network. The RoutingTableRecorder (*routingRecordes*) logs changes in the routing tables and interface tables of all hosts and routers. To complete the IPv4 network there is a Router module (*router*) that supports the Point-to-Point Protocol (PPP) to establish a direct connection between two endpoints and a StandardHost module (*server*) which provides protocols like TCP and UDP. The StandardHost acts also like the sender for the UDP packets.

The LTE Modules consist of eNodeB modules (*eNodeB*), the base stations for the LTE cellular network. Furthermore there is the LTEBinder module(*binder*), which stores the IP address in combination with an unique node id to tell this to a sender module. The PGN-GW module (*pgw*) is needed as connection between LTE and the Internet. The last LTE module to be considered is the LteChannelControl module (*channelControl*). It gets informed about the location of nodes and defines whether a node is within communication or interference distance. Depending on this information a nodes radio interface decides how to proceed with transmissions.

Vehicular Modules group the VeinsInetManager (*veinsManager*) and the Car module. These are modules provided by the Veins Framework and they are responsible for the interaction of SUMO with OMNeT++. The VeinsInetManager creates the interface to SUMO and manages the interaction of nodes which are represented as cars. These cars are then added as Car modules by the veinsManager at run time, therefore they are not visible in the image above. A more detailed description of the Car module is given in 4.3.2.

Figure 4.5 also shows the connections from the *server* to the *router* and from the *router* to the *pgw*. Both are connected via 10 Gigabit Ethernet, which looks in the network file as follows:

```
server.pppg++ <--> Eth10G <--> router.pppg++;
router.pppg++ <--> Eth10G <--> pgw.filterGate;
```

eNodeBs

The structure of an eNodeB with its submodules can be seen in 2.6. This subsection deals with embedding the base stations to position according to the OpenCellID database and creating the responsive connections. For this purpose the Python script named *createeNodefromDB.py* was written. It therefore queries the data from the previously created local mySQL database and creates a resulting .txt-file containing all eNodeBs located in the area of the Erlangen road network with the required connections for the .ned file. This is necessary because the area to be analyzed contains 66 eNodeBs. After running the script one can copy the provided information to the .ned file, which looks as follows:

```
eNodeB1: eNodeB { @display("p=251.240162,1464.765155;"); }
eNodeB2: eNodeB { @display("p=1112.901067,870.530046;"); }
...
eNodeB66: eNodeB { @display("p=1214.380900,333.056198;"); }

pgw.pppg++ <--> Eth10G <--> eNodeB1.ppp;
...
pgw.pppg++ <--> Eth10G <--> eNodeB66.ppp;

pgw.pppg++ <--> Eth10G <--> eNodeB1.ppp;
...
pgw.pppg++ <--> Eth10G <--> eNodeB66.ppp;

eNodeB1.x2++ <--> Eth10G <--> eNodeB22.x2++;
eNodeB1.x2++ <--> Eth10G <--> eNodeB41.x2++;
...
```

The eNodeBs contain their location information in the @display tag with the OMNeT++ coordinates. Each eNodeB also gets connected to the pgw and to all surrounding eNodeBs to provide LTE handover over the X2 interface. One eNodeB has exactly one communication with each other base station in a 1000m radius. The radius was set to this value to reduce runtime for setting the connections up during the simulation. Furthermore this radius is corresponds to a greater value than the highest range between two base stations without having another one in between.

Car Module

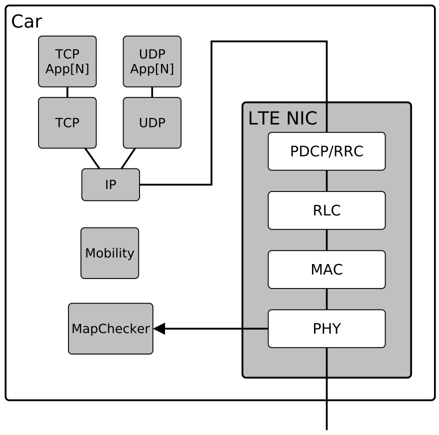

Figure 4.6: Car module with submodules.

The Car module is basically a UE with additional submodules. In addition to the modules known from Figure 2.6, the Mobility module and a MapChecker module are added for cars, which can be seen in Figure 4.6. The Mobility module is responsible for the position and orientation over time. It enables movement and is used by Veins to update the position of a car in the OMNeT++ simulation. The helper module MapChecker provides the logic for storing the coordinates. Therefore, every received UDP packet of a car which is defined as a message, gets checked by the *LteRealisticChannelModel* of the connected eNodeB. It is only transmitted to the cars Physical Layer, if no error occurred. The error-free messages will then be intercepted in the Physical Layer and send to the MapChecker. Whenever a message is received via the gate of the MapChecker, the position from the Mobility module is queried and stored in an array if it is at least 3 meters away from the previously stored coordinate. This has the reason not to make the arrays too large. When the simulation finishes, each car stores the values in a file named by its ID.

4.3.3 Simulation Initialization

Since the network structure with the individual modules is set, some relevant simulation-specific parameters are explained in this section. The simulation specific files containing the corresponding data are located in the Erlangen project which is a sub folder of the 'simulte_veins' workspace.

LTE Network

The first thing was to define the power provided by the eNodeBs to make the results comparable. Since a range of 500m was determined in 4.1.2 and the OpenCellID database does not contain this information about the power, a comparative value was determined by a simple simulation. The previously defined car module was used to move a vehicle linearly away from a microcell eNodeB while continuously receiving packets. The network itself consisted of exactly one omnidirectional eNodeB to avoid handover and signal interferences among others. The coordinates were stored if an error-free packet arrived as in the *ErlangenLTE* simulation. The used frequency and the channel model were the same as the ones from the ErlangenLTE simulation, except that the channel model did not include obstacle shadowing. After a few test runs with different transmission powers, a value of 0.1 W corresponding to 20 dBm was found, which on average had in the simulation area a range between 500m and 600m.

These determined values are used as parameters for the ErlangenLTE network to define the channel properties. Since the position of an eNodes is already provided by its *display* value in the network model, this information were transmitted to its Mobility module to enable position based calculations. In combination with the usage of band 7 from 2.3.4 and its frequency of 2.6 GHz and the omnidirectional microcell eNodeBs, the parameters define as follows:

```
**.channelControl.pMax = 0.1W
**.channelControl.carrierFrequency = 2600e+6Hz

*.eNodeB*.deployer.microCell = true
**.eNodeBTxPower = 20
**.txDirection = "OMNI"
**.eNodeB*.mobility.initFromDisplayString = true
```

The handover is enabled with a checking interval of 0.1s. Furthermore each eNodeB needs therefor a X2 application for each X2 connection. Since each eNodeB has to be as well server as client, each gets an unique server port and an unique client address. These values are also created by the Python script*createeNodefromDB.py* from 4.3.2, as it is depending on the number of eNodeBs and contains in case of the ErlangenLTE nework 765 client addresses. As these values are stored in the same file, one can copy them to the corresponding .ini file.

```
**.enableHandover = true
**.broadcastMessageInterval = 0.1s
*.eNodeB*.x2App[*].server.localPort = 5000 + ancestorIndex(1)

*.eNodeB1.numX2Apps = 14
*.eNodeB2.numX2Apps = 13
...

*.eNodeB1.x2App[0].client.connectAddress = "eNodeB22%x2ppp0"
*.eNodeB1.x2App[1].client.connectAddress = "eNodeB41%x2ppp0"
...
*.eNodeB23.x2App[0].client.connectAddress = "eNodeB2%x2ppp5"
*.eNodeB23.x2App[1].client.connectAddress = "eNodeB7%x2ppp6"
```

The Physical Layer channel model and the feedback parameters are set in a separate file called *config_chanel.xml*. It is provided by SimuLTE as an example of a realistic channel model. This file is used to define physical layer parameters like antenna gain, thermal noise and target Block Error Rate (BLER) and to enable the computation of shadowing, fading and cell interference. In addition, it sets the height of a building to 20m and the height of an eNodeB to 25m. The *config.xml* also contains sample parameters, but for the obstacle shadowing. These values come from the Veins framework which uses the VeinsInetManager (*veinsManager*) module to import the obstacles with their shapes from the *erlangen.poly.xml*.

```
**.lteNic.phy.channelModel=xmldoc("config_channel.xml")
**.feedbackComputation = xmldoc("config_channel.xml")

*.obstacles.obstacles = xmldoc("config.xml",
       "//AnalogueModel[@type='SimpleObstacleShadowing']/obstacles")
```

Cars

The cars are imported by the *veinsManager* and Traffic Control Interface (TraCI) from SUMO. Therefore the server needs to be started before the OMNeT++ simulation by running *sumo-launchd.py*. This provides the interface to SUMO which is accessible via *localhost:9999*. The update interval is set to the same value as the *positionUpdateInterval* and the step-length of the SUMO traffic simulation. The launch configuration for the SUMO simulation is defined in *erlangen.launchd.xml* including the path to the SUMO configuration file *erlangen.sumo.cfg*, *erlangen.poly.xml* containing building shapes, *erlangen.net.xml* containing the road network and *erlangen.rou.xml* containing the vehicles with their routes. The dynamic cell association will also be enabled for the cars, which initiates a handover if another eNodeB provides a better signal quality based on the SINR value.

```
*.veinsManager.moduleType = "lte.corenetwork.nodes.cars.Car"
*.veinsManager.moduleName = "car"
*.veinsManager.launchConfig = xmldoc("erlangen.launchd.xml")
*.veinsManager.updateInterval = 0.01s
*.veinsManager.port = 9999
*.veinsManager.host = "localhost"

**.deployer.positionUpdateInterval = 0.01s
**.dynamicCellAssociation = true
```

Vehicle Routes

The initial idea was to use DUAROUTER to define random routes, which are checked for full edge coverage in the road network. This led to a route file consisting of 50 vehicles, but had the disadvantage that these had to cover very long distances and Edges were used multiple times. This led to an extremely long run time of the simulation, so the following different implementation was chosen. This is realized in the Python script called *edgeToRoutes.py*, that creates exactly one vehicle for each edge of the Erlangen road network. Since OMNeT++ is a discrete event based simualtion framework, the vehicles travel the edges by starting one after the other to avoid delays caused by other vehicles. The vehicles and the corresponding routes are then stored in the SUMO specific file named *erlangen.rou.xml*.

UDP Application

As mentioned before an UDP based application is used to send data packets from the server to the cars for the availability measurement. The Voice over Internet Protocol (VoIP) is used for this purpose to continuously send packets to the cars. The VoIP application is provided by the SimuLTE framework and configured by setting the server as the VoIP sender with 457 udp applications, one for each car as VoIP receiver. Each VoIP receiver obtains its individual address and port defined at the VoIP server. With a little buffer of 0.05 s the server starts sending packets to the cars until they left the simulation.

```
*.server.numUdpApps = 457
*.server.udpApp[*].typename = "VoIPSender"
*.server.udpApp[*].localPort = 3000 + ancestorIndex(0)
*.server.udpApp[*].destAddress = "car[" + string(ancestorIndex(0)) + "]"
*.server.udpApp[*].startingTime = 0.05s

*.car[*].numUdpApps = 1
*.car[*].udpApp[0].typename = "VoIPReceiver"
```

4.3.4 Road Network Adaption

The final step is to create the resulting SUMO road network file based on the resulting coordinates. This is realized by the Python script called *simulativeRoadNetworkAdaption.py*. This uses, just like in the Mathematical Realization, the file handler to open the road network XML file and to save the corresponding modifications in a new file called *simulativeAvailable.net.xml*. The decision whether an edge is available or not, is done by iterating over all edges of the road network and check the corresponding coordinates stored by the cars during the simulation. A road is blocked if a coordinate is more than 10 meters away from the previous one. This value was chosen because a coordinate, as defined in the OMNeT++ simulation, is at least 3 meters away from previous and a thus a buffer was also considered. For the Simulative Approach based road network the edges are disallowed for vehicles of type *custom2*, which does also corresponds to teleoperated vehicles, but allows the representation of both approaches in one common road network.

Figure 4.7: Resulting Route Network of the Simulative Approach visualized in NETEDIT.

The resulting road network based on the Simulative Approach can be seen in Figure 4.7. The black colored edges are available routes, which can be used by teleoperated as well as normal vehicles. The red edges, on the other hand, are outside of the signal area and thus not accessible for teleoperated vehicles.

5 Evaluation and Results

This chapter presents the evaluation of the road network adaption processes with their results. Furthermore the effects of the resulting road networks of the different approaches are compared with a conventional vehicle.

5.1 Road Network Adaption Processes

The processes of the two approaches differ mainly in the runtime and the resulting road network. The runtime for the creation of the resulting adapted road network based on the Mathematical Approach is only a few seconds. Since the result depends only of the approximate range of a base stations signal without further interfering factors, the blocked roads of the resulting road network displayed in Figure 4.4 correspond exactly to the grey area of the coverage map displayed in the right image of Figure 3.2.

The execution of the process based on the Simulative Approach required much more time than the mathematical one. While the creation of the road network file with the help of the determined coordinates took only a few seconds, the simulation to generate these coordinates took about 18 hours (63942 s) on a Linux based computer with an Intel(R) Core(TM) i5-2410M processor and a clockspeed of 2.3 GHz. The high runtime is mainly due to the number of eNodeBs (66 eNodeBs) and the resulting high number of X2 applications, which constantly calculates the SINR value to initiate a handover. The resulting road network of the Simulative Approach, displayed in Figure 4.7, looks slightly more limited for teleoperated vehicles. The blocked roads are the result of a lack of network availability, which is displayed in Figure 5.1. There one can see the number of error-free received UDP packets at simulation time. The blue dots represent *Car 1* which continuously received packets without any interruptions. This can be identified by the straight and lined up arrangement.

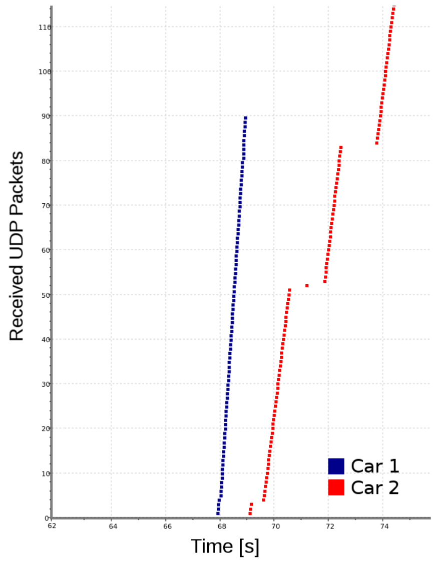

Figure 5.1: Received UDP packets of different test cars.

Car 2 on the other hand represents a vehicle which received UDP packets with interruptions. As one can see in Figure 5.2 this vehicle checked the road in the zoomed area, where the same colored red cross is set. The road to be inspected is located between some buildings, represented as light green rectangles, which leads to short-term interruptions during transmission. Due to the fact that no packet was received at least once over a period of 1 second. At the maximum speed of 13.89 m/s, this corresponds to a failure of more than 10 m leading to a road being blocked for a teleoperated vehicle.

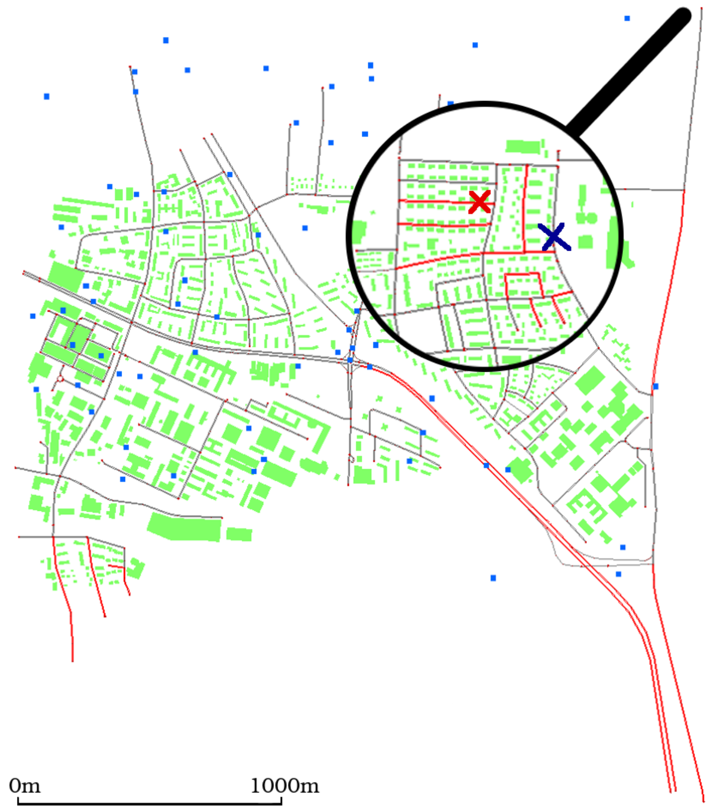

Figure 5.2: Resulting route network of the Simulative Approach visualized in NETEDIT with eNodeBs (blue) and buildings (green).

Furthermore, the resulting road network from Figure 5.2 shows that roads with high maximum speeds are susceptible to the transmission of packets. Despite the nearby eNodeBs, which are represented by small blue squares, the required bandwidth cannot be provided on these roads.

5.2 Road Network Comparison

The restricted road networks as basis for route determination for teleoperated vehicles were analyzed in the previous section on the basis of their creation process and their resulting characteristics. In this section a SUMO test scenario was created for performing route determination based on the resulting road networks and the default network to compare them with each other.

5.2.1 Evaluation Setup

For the evaluation scenario, a SUMO road network file was first created, which represents the entire road network as well as the restricted road networks of both approaches. In the Mathematical Realization, roads not covered by the Cellular Network were blocked in the road network file for vehicles of type 'custom1' and in the Simulative Realization for vehicles of type 'custom2'. By merging the networks, the comparison road network file was created, which contains roads blocked for different vehicle types. This means that different vehicles can be used to access individual road networks: type 'default' for the default, unrestricted network, 'custom1' for the mathematically restricted network and 'custom2' for the simulative restricted network. With the help of SUMO's DUAROUTER tool, randomly 26 different start and finish edge pairs were determined called *flows*. Then one of each of the three previously mentioned vehicle types was placed on the starting edge. SUMO internally calculated and simulated the shortest path to the target edge for these vehicles.

5.2.2 Results

The output file created by SUMO stored the results of the simulated scenario with information about the corresponding departure time, the arrival time, the duration and the route length among others. Since the duration is not only dependent on the road network, but also on the traffic, the route length was chosen as comparison factor. The resulting route lengths of each flow is represented in the bar chart displayed in Figure 5.3. The not displayed bars of custom1 and custom2 at flow 5 and 7, mean that either no route could be found for the respective vehicle types, or that the start or end edge blocked for this type of vehicle. For these flows, this would mean that no teleoperated vehicle would be able to get from the start edge to the target edge, as it has to be driven through uncovered areas. Furthermore, one can see that for example on flow 19,20 and 21 the route was longer, what means that a teleoperated vehicle with route determination based on the simulative road network had to make a detour to reach the destination without leaving the cellular network covered area. At flow 9, one can see that both, the 'custom1' and the 'custom2' typed vehicle had to make a detour. This implies that there is a part of the road network blocked for both vehicles.

Based on the simulative restricted road network and without considering flow 5 and 11, a teleoperated vehicle had to accept a detour of approx. 6.5% comparable to a conventional vehicle.

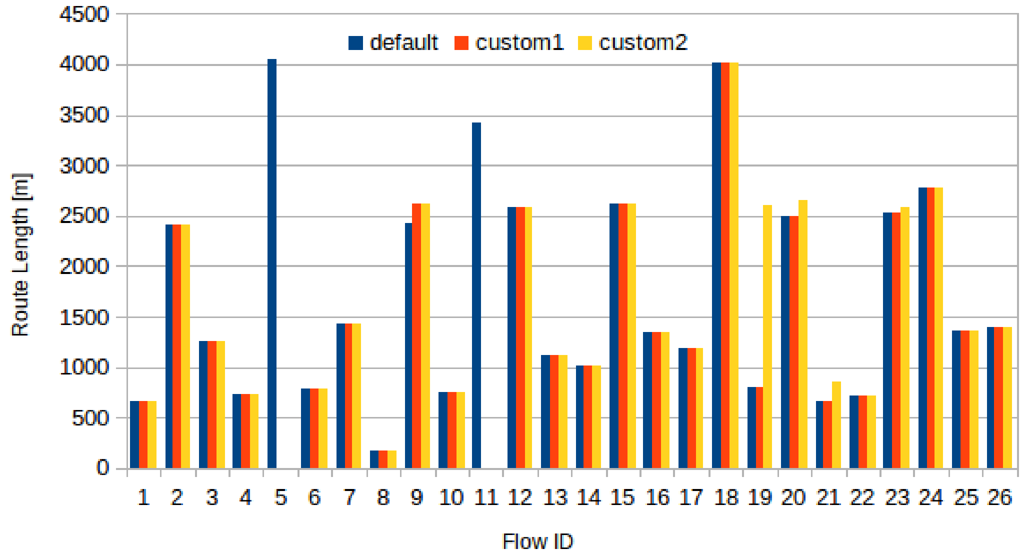

Figure 5.3: Route length for each flow and each vehicle type representing the default, mathematical and the simulative road network.

Based on the output data of the comparison scenario and on the resulting restricted road networks of both approaches, one can say that at least each blocked road from the Mathematical Approach road network is also blocked in the Simulative Approach road network.

6 Conclusion and Discussions

6.1 Conclusion

How can cellular network availability be projected to a digital road network to enable route determination for teleoperated driving?

Two different approaches were designed and implemented to determine the coverage of the cellular network in a road network. Both take a road network as a basis and determine the areas in which there is no coverage of the cellular network. The roads within this area are then blocked for a resulting road network, which can be used as a basis for conventional route determination. The road network of Erlangen was restricted in the Mathematical Approach by simply blocking roads out of a provided approximated range. In the simulative approach, the cellular network coverage was determined by combination of network and vehicle traffic simulation. The resulting road network is determined with less computational effort and therefore very fast.

The Simulative Approach, on the other hand, considers several interfering factors that can affect the network. This results in a comparatively long execution time for the simulative determination of the availability of the cellular network in the road network.

In summary, an interaction of both approaches can produce the result of the simulative approach in a shorter time. By first applying the Mathematical Approach with a generous signal range as a pre-selection. This can reduce the number of roads to be considered for the Simulative Approach which could reduce the runtime of the simulation.

6.2 Limitations

The results of this thesis come with several limitations. Since the exact information about mobile masts such as their position and exact performance is not publicly available, one is dependent on open-source databases. Since the exact information about cell towers such as their position and exact performance is not publicly available, one has to rely on open-source databases. However, these are mainly the result of smartphones that share location and mobile network information. The database then forms an average value and a corresponding range.

6.3 Future work

Based on the Simulative Approach shown in this thesis for creating a restricted road network for teleoperated vehicles one can adapt and extend the simulation to create Scenario on a bigger road network with multiple teleoperated vehicles. Among other things, the network utilization rate could be analyzed in this context. This then offers the possibilities to realize the Dynamic Approach for real-time route determination based on traffic and network utilization as described in **conceptsGraph** The 5G network offers further possibilities. The approaches for 5G could be adapted in the future as soon as the infrastructure is up. Since both the simulation and, above all, the lack of information about cell towers, this offers a large scope for approaches for future work.

List of Figures

List of Tables

Bibliography

[1] S. Neumeier, N. Gay, C. Dannheim, and C. Facchi, "On the way to autonomous vehicles teleoperated driving," in *AmE 2018-Automotive meets Electronics; 9th GMM-Symposium*, VDE, 2018, pp. 1–6.

[2] D. Watzenig and M. Horn, *Automated Driving: Safer and More Efficient Future Driving*. Springer International Publishing, 2016, ch. Introduction to Automated Driving, pp. 1–16, ISBN: 9783319318950.

[3] J. Dokic, B. Müller, and G. Meyer, "European roadmap smart systems for automated driving," *European Technology Platform on Smart Systems Integration*, 2015.

[4] *Audi traffic jam pilot*, `https://www.audi-mediacenter.com/en/press-releases/automated-driving-at-a-new-level-the-audi-ai-traffic-jam-pilot-9300`, Accessed: 2019-03-17.

[5] F. Diermeyer, S. Gnatzig, F. Chucholowski, T. Tang, and M. Lienkamp, "Der mensch als sensor-der weg zum teleoperierten fahren," in *AAET-Automatisierungssysteme, Assistenzsysteme und eingebettete Systeme für Transportmittel*, 2011, pp. 119–135.

[6] A. F. Winfield, "Future directions in tele-operated robotics," *Telerobotic applications*, pp. 147–163, 2000.

[7] A. Hosseini and M. Lienkamp, "Predictive safety based on track-before-detect for teleoperated driving through communication time delay," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2016, pp. 165–172.

[8] S. Gnatzig, F. Chucholowski, T. Tang, and M. Lienkamp, "A system design for teleoperated road vehicles.," in *ICINCO (2)*, 2013, pp. 231–238.

[9] H.-J. Negele, "Anwendungsgerechte konzipierung von fahrsimulatoren für die fahrzeugentwicklung," PhD thesis, Technische Universität München, 2007.

[10] A. Hosseini and M. Lienkamp, "Enhancing telepresence during the teleoperation of road vehicles using hmd-based mixed reality," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2016, pp. 1366–1373.

[11] A. Burkert, *Teleoperiertes autofahren–eine chance für die elektromobilität*, `https://www.springerprofessional.de/automobil---motoren/teleoperiertes-autofahren-eine-chance-fuer-die-elektromobilitaet/6590738`, Accessed: 2018-12-17, 2013.

[12] M. Sauter, *Grundkurs Mobile Kommunikationssysteme: LTE-Advanced Pro, UMTS, HSPA, GSM, GPRS, Wireless LAN und Bluetooth*. Springer Fachmedien Wiesbaden, 2018, p. 76.

[13] A. R. Mishra, *Advanced cellular network planning and optimisation: 2G/2.5 G/3G... evolution to 4G*. John Wiley & Sons, 2007.

[14] M. R. Salehifar, S. Soleimany, and H. Karbalaee, "Bss parameters and their influences in gsm mobile networks," in *2012 6th International Conference on Application of Information and Communication Technologies (AICT)*, IEEE, 2012, pp. 1–4.

[15] G. Gu and G. Peng, "The survey of gsm wireless communication system," in *2010 international conference on computer and information application*, IEEE, 2010, pp. 121–124.

[16] C. Ball, K. Ivanov, and F. Treml, "Contrasting gprs and edge over tcp/ip on bcch and nonbcch carriers," in *2003 IEEE 58th Vehicular Technology Conference. VTC 2003-Fall (IEEE Cat. No. 03CH37484)*, IEEE, vol. 4, 2003, pp. 2564–2568.

[17] S.-L. Tsao, "Improving packet delivery in gprs/umts networks," in *The 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, IEEE, vol. 5, 2002, pp. 2024–2028.

[18] W.-Y. Chen, J.-L. Wu, and H.-H. Liu, "Performance analysis of radio resource allocation in gsm/gprs networks," in *Proceedings IEEE 56th Vehicular Technology Conference*, IEEE, vol. 3, 2002, pp. 1461–1465.

[19] F. Gleissner and S. Hanus, "The comparison of gsm-edge signal transmission over awgn and rayleigh fading channel," in *2008 18th International Conference Radioelektronika*, IEEE, 2008, pp. 1–4.

[20] D. Molkdar, W. Featherstone, and S. Larnbotharan, "An overview of egprs: The packet data component of edge," *Electronics & Communication Engineering Journal*, vol. 14, no. 1, pp. 21–38, 2002.

[21] M. Zivkovic and Y. Wang, "Qos attributes for packet switched services in 3rd generation mobile systems (umts)," 2004.

[22] W. Wong and E. S. Sousa, "Feasibility study of tdd-and fdd-cdma frequency sharing cellular networks," in *ICUPC'98. IEEE 1998 International Conference on Universal Personal Communications. Conference Proceedings (Cat. No. 98TH8384)*, IEEE, vol. 1, 1998, pp. 531–535.

[23] *The mobile broadband standard*, `https://www.3gpp.org/About-3GPP`, Accessed: 2019-01-09.

[24] H. G. Myung, "Technical overview of 3gpp lte," *Polytechnic University of New York*, 2008.

[25] Z. Shen, A. Papasakellariou, J. Montojo, D. Gerstenberger, and F. Xu, "Overview of 3gpp lte-advanced carrier aggregation for 4g wireless communications," *IEEE Communications Magazine*, vol. 50, no. 2, pp. 122–130, 2012.

[26] L. Lei, Z. Zhong, C. Lin, and X. Shen, "Operator controlled device-to-device communications in lte-advanced networks," *IEEE Wireless Communications*, vol. 19, no. 3, pp. 96–104, 2012.

[27] P. Beming, L. Frid, G. Hall, P. Malm, T. Noren, M. Olsson, and G. Rune, "Lte-sae architecture and performance," *Ericsson Review*, vol. 3, pp. 98–104, 2007.

[28] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. Soong, and J. C. Zhang, "What will 5g be?" *IEEE Journal on selected areas in communications*, vol. 32, no. 6, pp. 1065–1082, 2014.

[29] W. Xiang, K. Zheng, and X. S. Shen, *5G mobile communications*. Springer, 2016.

[30] T. Chamberlain, *Learning OMNeT++*, ser. Community experience distilled. Packt Publishing, 2013.

[31] A. Varga and R. Hornig, "An overview of the omnet++ simulation environment," in *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops*, ser. Simutools '08, Marseille, France: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, 60:1–60:10, ISBN: 978-963-9799-20-2.

[32] A. Varga, "Discrete event simulation system," in *Proc. of the European Simulation Multiconference (ESM'2001)*, 2001.

[33] *Omnet - simulation manual*, `https://doc.omnetpp.org/omnetpp/manual/`, Accessed: 2019-02-11.

[34] *Inet framework*, `https://inet.omnetpp.org/`, Accessed: 2019-02-07.

[35] A. Yonis, M. F. L. Abdullah, and M. Ghanim, "Lte-fdd and lte-tdd for cellular communications," *Progress in Electromagnetics Research Symposium*, Jan. 2012.

[36] A. Virdis, G. Stea, and G. Nardini, "Simulating lte/lte-advanced networks with simulte," in *Simulation and Modeling Methodologies, Technologies and Applications*, Springer, 2015, pp. 83–105.

[37] G. Nardini, A. Virdis, and G. Stea, "Simulating device-to-device communications in omnet++ with simulte: Scenarios and configurations," *ArXiv preprint arXiv:1609.05173*, 2016.

[38] G. Nardini, A. Virdis, and G. Stea, "Modeling x2 backhauling for lte-advanced and assessing its effect on comp coordinated scheduling," in *2016 1st International Workshop on Link- and System Level Simulations (IWSLS)*, Jul. 2016, pp. 1–6.

[39] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of sumo-simulation of urban mobility," *International Journal On Advances in Systems and Measurements*, vol. 5, no. 3&4, 2012.

[40] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using sumo," in *The 21st IEEE International Conference on Intelligent Transportation Systems*, IEEE, 2018.

[41] OpenStreetMap contributors, *Planet dump retrieved from https://planet.osm.org*, `https://www.openstreetmap.org`, Accessed: 2019-01-09, 2017.

[42] *Opendrive - home*, `http://www.opendrive.org/`, Accessed: 2019-02-20.

[43] *Netedit*, `https://sumo.dlr.de/wiki/NETEDIT`, Accessed: 2019-02-07.

[44] *Sumo - simulation/routing*, `https://sumo.dlr.de/wiki/Simulation/Routing`, Accessed: 2019-02-28.

[45] C. Sommer, R. German, and F. Dressler, "Bidirectionally coupled network and road traffic simulation for improved ivc analysis," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, Jan. 2011. DOI: `10.1109/TMC.2010.133`.

[46] D. Jungnickel, *Graphen, Netzwerke und Algorithmen*. Bibliographisches Institut, 1987, p. 3.

[47] K. Mehlhorn and P. Sanders, "Algorithms and data structures: The basic toolbox," in. Springer, 2008, ch. Shortest Paths, 191ff.

[48] D. Delling, P. Sanders, D. Schultes, and D. Wagner, "Engineering route planning algorithms," in *Algorithmics of large and complex networks*, Springer, 2009, pp. 117–139.

[49] T. Ottmann and P. Widmayer, "Algorithmen und datenstrukturen," in. Springer Berlin Heidelberg, 2017, ch. Dijkstras Algorithmus zur Berechnung kürzester Wege, pp. 405–408.

[50] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, Jul. 1968.

[51] R. Geisberger, P. Sanders, D. Schultes, and D. Delling, "Contraction hierarchies: Faster and simpler hierarchical routing in road networks," in *International Workshop on Experimental and Efficient Algorithms*, Springer, 2008, pp. 319–333.

[52] R. Güting and S. Dieker, "Datenstrukturen und algorithmen," in. Springer Fachmedien Wiesbaden, 2018, ch. Graph-Algorithmen, pp. 213–248.

[53] V. Demary and C. Rusche, "Zukunftsfaktor 5g: Eine ökonomische perspektive," IW-Report, Tech. Rep., 2018.

[54] *Menu map view - opencellid wiki*, `http://wiki.opencellid.org/wiki/Menu_map_view`, Accessed: 2019-03-05.

[55] *Gnu sed*, `https://www.gnu.org/software/sed/`, Accessed: 2019-03-05.

[56] M. Widenius and D. Axmark, *Mysql Reference Manual*, 1st, P. DuBois, Ed. Sebastopol, CA, USA: O'Reilly & Associates, Inc., 2002.

[57] *Integrating simulte and veins*, `http://simulte.com/add_veins.html`, Accessed: 2019-03-10.