Delft University of Technology
Master of Science Thesis in Embedded Systems

Reducing journey times for en-route charging using V2X communication

Sebastiaan Kruit

Embedded Systems

TU Delft Delft University of Technology

Reducing journey times for en-route charging using V2X communication

Master of Science Thesis in Embedded Systems

Embedded Systems Group
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
Van Mourik Broekmanweg 6, 2628 XE Delft, The Netherlands

Sebastiaan Kruit

22nd of February 2024

Author

Sebastiaan Kruit

Title

Reducing journey times for en-route charging using V2X communication

MSc Presentation Date

1st of March 2024

Graduation Committee

Prof. dr. Koen Langendoen (chairman)	Delft University of Technology
Prof. dr. Mathijs de Weerdt	Delft University of Technology
Pasqual Boehmsdorff	Porsche Engineering Services GmbH

Abstract

This thesis project explores how V2X communication between electric vehicles (EV) and charging stations can be used to reduce en-route charging times. This is done using standardized V2X messages for EV charging, as well as proposing an extension to these messages to include data on the intentions of other vehicles. As individual vehicles can have significant effects on the total waiting time at EV charging stations, knowing the intentions of other vehicles will allow drivers to better avoid congested charging stations and achieve a lower total journey time. The performance of the system is evaluated using state-of-the-art traffic and communications simulators showing that using V2X can reduce journey times by 70%. Aside from demonstrating the performance using simulation the system has also been implemented on a real life test vehicle using functional V2X hardware to show that the system is viable for implementation.

Preface

I started my BSc degree in Computer Science and Engineering because I had an interest in programming and creating things. During my gap year at Forze Hydrogen Racing I learned about writing software for embedded systems, which I enjoyed a lot. This led to starting the a Master in Embedded Systems at the TU Delft. For my thesis project I wanted to combine working in the industry with living abroad. The project at Porsche Engineering was perfectly suited for this. I really enjoyed my time in Germany, everything I learned there, and the people I met. I would like to thank my supervisor from the TU Delft, Koen Langendoen, for his help with structuring the project and giving valuable feedback on the thesis report. Secondly, I want to thank Pasqual Boehmsdorff for the opportunity of writing my thesis at Porsche Engineering, and Sai Praneeth Reddy Animireddy for the day-to-day guidance and technical support with the project. Additionally, I want to thank Emsal Kaya, Yoshua Neuhard, Vaishnav Kolamudi and Vijay Mistry for the great times in, and outside of the office. Finally, I want to thank my family for supporting me on this adventure.

Put acknowledgments here.

Sebastiaan Kruit

Delft, The Netherlands
22nd February 2024

Contents

Chapter 1

Introduction

Vehicle-to-Everything (V2X) is an advanced communication technology that allows vehicles to communicate with other entities that have an impact on traffic. The V2X ecosystem consists of multiple different communication links, including Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), Vehicle-to-Cloud/Network (V2C/V2N) and Vehicle-to-Pedestrian (V2P). Most of which also function in reverse direction. The main use for V2X communication is the implementation of Intelligent Transport Systems (ITS). Examples of Day 1 use cases of V2X based ITS are Green Light speed recommendations, Traffic Jam warnings, Left-Turn assist and Emergency Brake Warning. While advanced use cases include remote driving and platooning [4]. Today, there are two radio access technologies (RATs) that allow for V2X communication. These are Dedicated Short Range Communications (DSRC) and Cellular-V2X (C-V2X). DSRC is based on the IEEE 802.11p standard, while C-V2X is developed by the 3rd Generation Parnership Project (3GPP) and based on the Long Term Evolution (LTE) standard.

Development of V2X based ITS started in 2010 with the publication of the IEEE 802.11p standard [29]. In 2016, Toyota brought the first vehicles to the Japanese market that were equipped with DRSC based V2X technology, followed by Cadillac in the US in 2017 and Volkswagen with the 2020 Golf 8 in Europe. The development of C-V2X started in 2016 with its inclusion in 3GPP Release 14 [26]. The main benefits of C-V2X over DSRC are that it leverages the already widespread cellular infrastructure and that it allows for multiple devices to transmit at the same time in a given channel [59], which increases performance and reliability especially when vehicle density is high. In addition to uplink-downlink communication, which depends on cellular infrastructure, C-V2X also defines a sidelink channel (PC5) that allows direct communication between entities to support low-latency communication. Communication through the PC5 interface can be utilized in two modes: mode 3, where resources allocation of the sidelink channel is managed by a base station; and mode 4, where vehicles manage resource allocation themselves in the case that cellular coverage is not available. The different communication modes of C-V2X are shown in Figure 1.1. According to studies, C-V2X provides significant performance advantages over DSRC as it has a bigger transmission range in both Line-of-Sight (LOS) and Non-Line-of-Sight (NLOS) conditions as well dealing better with interference [2].

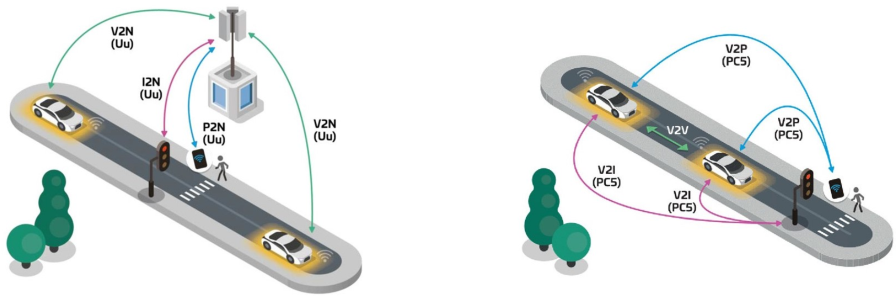

Figure 1.1: **C-V2X communication modes. Left: entities communicate with uplink (UL) and downlink (DL) channels using a base station; Right: entities use the PC5 interface to communicate directly without requiring a cellular network. [3]**

Even though there are differences in performance, both technologies are shown to be adequate for implementation of basic safety systems that require a message frequency of 10Hz as long as vehicle density is limited [48]. However, for more advanced systems such as platooning and remote driving with much higher Quality of Service (QoS) requirements [1] both technologies are insufficient. To support these systems in the future both IEEE and 3GPP are working on evolutions of these existing technologies, IEEE 802.11bd and NR-V2X respectively, which should be able to deliver an end-to-end latency of 3ms and a reliability of 99.999% for the most demanding use cases.

Another big development in the automotive industry is the adoption of electric vehicles (EVs). The world is currently at a key point regarding climate change. If big changes to emissions are not made soon climate change could become irreversible. According to EU studies, passenger cars make up around 12% of total emissions of CO_2 [14], which is why the European Comission has recently passed legislation saying that in 2035 all new cars should have zero emissions [27]. A big role in this transition is reserved for EVs. In 2022, 14% of all new sold cars were electric. This number is expected to increase to 18% in 2023, when EVs should account for over 10% of road vehicles according to predictions based on current policies [36]. However, the adoption of EVs by the general public faces some big barriers, among which are problems regarding range, charging infrastructure and charging duration [40]. These concerns are commonly referred to as "Range Anxiety". The industry is hard at work tackling these issues, which can be seen by an exponential growth in battery capacity of newer vehicles [57], and fast growth in both the number of available charging ports as well as the power output of charging ports [10]. However, these measures are not always effective in resolving the emotional concerns regarding range anxiety as seen in studies showing 85-90% of trips could already covered by current EVs [47], while 84% of people still report range anxiety deterring them from making a trip with an EV [42].

At Porsche Engineering, in the Connectivity department, engineers are working hard to develop V2X functions using the C-V2X standards. On top of the use cases specified by organisations as the 5G Automotive Association (5GAA) [4], functions are being developed that improve the user-experience while driving their car alone; or in groups. This thesis was commissioned by Porsche Engineering with the goal of exploring possible V2X applications related to EV charging stations. This is one of many fields in which the team sees potential of using V2X to improve Porsche premium vehicles. The people at Porsche Engineering truly see V2X as an important part of tomorrows' vehicles, and have set up multiple projects, including this thesis, as well as large-scale tests at the Nardò Technical Center (NTC) [53]. The main requirements for the thesis project were to develop an application using V2X with emphasis on the vehicle behavior, and to provide a demonstrative showcase using a provided test vehicle.

This thesis utilizes V2X messages defined by the European Telecommunications Standards Institute (ETSI), specifically focusing on the electric vehicle charging spot notification (EVCSN) [22] and the electric vehicle recharging spot reservation (EV-RSR) procedure [23], as a baseline. The objective is to propose various strategies for selecting charging stations for en-route charging. Additionally, an extension is proposed to the EVCSN message to include vehicles arriving at the charging station in the near future, enhancing the ease with which vehicle can select the optimal charging station. The performance of the charging station selection strategies is evaluated through a comprehensive large-scale traffic simulation. Furthermore, the systems' practical viability for real-world implementation is demonstrated by integrating the simulation with the Jupiter [32] test vehicle, which is equipped with V2X hardware.

1.1 Objectives

The objective of this thesis project is to improve the experience of Porsche users by implementing an application using V2X in the context of EV charging. This will be achieved by setting up a simulation, which can be used to evaluate different strategies for charging station selection for en-route charging. These strategies can then be compared to each using using following metrics: waiting time at the charging station and total journey time from origin to destination. The simulation should be capable of simulating the transmission of V2X messages, which allows the vehicles to communicate with each other, and the charging stations. This communication should play a key role in the evaluated strategies, and should be protocols used should be based on the European V2X standards published by ETSI. Finally, the used messages should be able to be transmitted and received by existing V2X capable hardware, such that the strategies used in the simulation can easily transition to applications on actual vehicles.

1.2 Challenges

At the start of the thesis project the tools available to develop and evaluate V2X application at Porsche Engineering consisted of two sets of C-V2X capable hardware kits, a virtual environment able to emulate the behavior of those hardware kits, and a test vehicle equipped with an Linux-based computing system to which a hardware kit could be connected. This setup is suitable for building and testing applications consisting of one or two vehicles. However, to achieve the goals outline in Section 1.1 a setup is required that can handle large amounts of vehicles. In addition to that the software provided with the hardware kits only supported a limited set of ETSI defined V2X messages, which did not include the EV-RSR messages, and only the currently proposed version of the EVCSN message. During this thesis work was done to create a new set of tools for the creation and evaluation of V2X applications. The main challenges encountered in this process can be summarized as follows:

- Setting up of a simulation environment capable of handling large numbers of vehicles, support electric vehicles and charging stations, simulation of V2X communication, and the ability the create applications to influence the behavior of entities in the simulation.
- Modify the software stack on the C-V2X hardware kits to allow sending/receiving of ETSI standardized messages related to EV charging, not included in the provided implementation.
- Integration of an existing test vehicle with the simulation to be able to demonstrate V2X applications with large numbers of vehicles without requiring big fleet of test vehicles and hardware.

1.3 Contributions

The main contributions of this work can be summarized as follows:

- Setup of a large-scale traffic simulation at Porsche capable of simulating large numbers of vehicles as well as the transmission of V2X messages.
- Integration of the simulation with existing V2X capable hardware to establish a connection between the simulation and a real-life test vehicle.
- Usage of the EV-RSR procedure to communicate a vehicles' charging intention with charging stations, without the use of blocking reservations, enabling charging stations to have a more accurate knowledge of their load demands.
- An extension to the current standard of the EVCSN message to include vehicles arriving at the charging station in the near future in addition to the current waiting times, giving vehicles more information on the state of the charging station.

1.4 Structure

The rest of this thesis is structured as follows: Chapter 2 discusses research done in the fields of EV charging and charging station selection. In Chapter 3 the architecture of the simulation and the different strategies for charging station selection is given, and in Chapter 4 the implementation details for the system are given. Chapter 5 explains the simulations that were performed to evaluate the performance of the different strategies and discusses the obtained results. Finally, in Chapter 6 the work done during this thesis project is summarized, and Chapter 7 gives some recommendations for how this simulation can be extended in the future, and the research that can be done to improve en-route charging even further.

Chapter 2

Related Work

Even though EVs were among some of the first cars built, they have only started to gain traction again in recent years due to their role in fighting climate change and reaching goals for future emission levels. Because of this, research into EVs, and various associated aspects are currently very active fields. In this chapter, research into the fields of EV charging, charging station selection, and the simulation of EVs are discussed. In doing so the current state-of-the-art is introduced, and the role this thesis work can play in extending that is shown.

2.1 EV Charging

Charging of EVs has a significant impact on the overall load on the electrical grid since the battery capacity of a modern EV is multiple times higher than the daily electricity usage of an average household. As both the supply and demand of electricity are not constant throughout the day, researcher are looking into smart solutions for better matching supply and demand, since energy storage and up-/down-ramping of production are both costly. The term used for an electrical grid that uses sensors and software to better match supply and demand is a smart grid.

Integration of EVs into the smart grid can have substantial benefits for infrastructure costs of the electrical grid. Implementation of smart charging strategies can reduce the required investment costs into power distribution infrastructure by 60-70% compared to uncontrolled charging, as reported in [25]. These smart charging strategies can take multiple different forms.

A basic version of a smart charging strategy is to simply charge vehicles during off-peak hours. Energy demand is shown to have predictable peaks based on season and time of day in [9, 34, 63]. By preponing or postponing charging to off-peak hours based on the predicted peaks [54] reports that it is possible to prevent peak loads on the electrical grid increasing with EV penetration rates of up to 20%, which is twice as high as what has currently been achieved [36]. Minimizing the increase in peak loads is crucial, as it means less investments have to be made to upgrade the existing electrical infrastructure.

More advanced strategies include monitoring real-time electricity rates using uni-directional communication from the smart grid to the charging equipment. The charging equipment then decides at what times to charge using historical

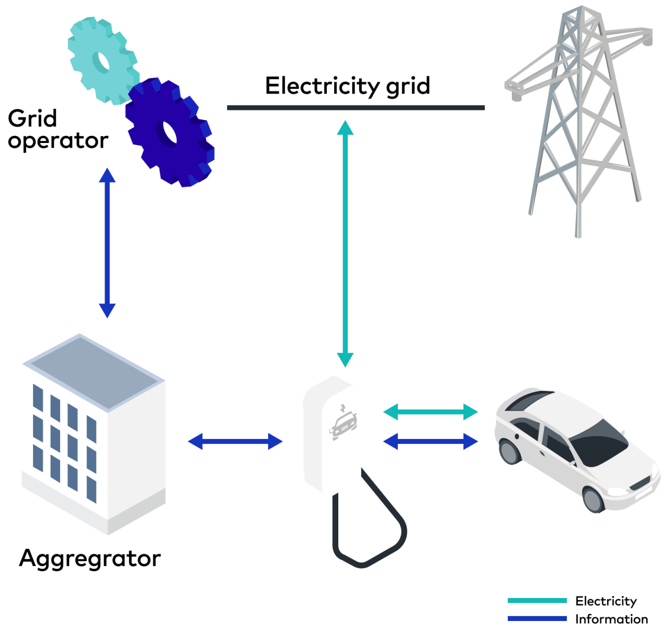

Figure 2.1: **Flows of energy and information in an aggregator managed V2G scenario. Showing communication lines between vehicle and charging station, charging station and aggregator, and aggregator and grid operator [43].**

price data or certain price thresholds. According to [54] this results in a better distribution of the energy demand over the course of the day. However, if every vehicle tries to charge during the time of minimal charging costs this can actually have adverse effects as shown in [46]. Instead the authors of that paper propose a method that includes a distribution system operator (DSO) that calculates the charging power available to each vehicle based on load forecasts as to not overload the grid. The DSO entity is often referred to as an EV aggregator, which is a non physical entity that manages the charging demands of a large group of EVs. These systems reduce the number of communication links the grid operators have to manage, reducing the investments that need to be made to build and upkeep that infrastructure and instead leaving more resources for the electrical grid itself. Aggregators can be implemented using different strategies, tailored to achieving specific goals. Their use shows promising results as reported in [13, 28, 30, 64, 69].

A different area of research focuses on an upcoming technology in the EV charging field: Vehicle-to-Grid (V2G). Most EVs tend to be unused for the majority of the day, and aside from recharging the battery to its full capacity in this time its idle. However, during this time the battery can be used to help the power grid in balancing its voltage and load. V2G depends on EVs equipped with bi-directional charging technology to allow grid balancing not only by modulating charging rates but also through discharging the stored energy in the battery back to the grid. To reach the climate goals, investments into renewable energy has increased, with Germany aiming for 50% of power to be from renewable sources in 2023 [61]. However, with renewable energy sources also come problems from fluctuations in production as shown in [11]. Dealing with these fluctuations properly will require the usage of energy storage devices as discussed in [66], with the need for storage devices increasing with the growing market share of renewable energy.

Instead of depending on investments into energy storage facilities the increasing popularity of EVs can be used to assist in this issue. V2G allows EVs to function as flexible energy storage whenever they are not in use. Generally, there are two different scales at which V2G can be implemented. The first is connecting the vehicle only to a home, sometimes referred to as Vehicle-to-Home (V2H). This allows homeowners to better manage their energy production and usage and increases their independence of the power grid. The second implementation is to allow the vehicle to connect to the power grid to exchange energy. An example of the flow of information and power in such an implementation is shown in Figure 2.1. Different control strategies for V2G are researched in [6, 31, 68]. The authors of [31] discuss different control schemes for V2G as well as problems that need to be addressed before it is able to be implemented on a larger scale, recommending focus on motivating EV drivers to participate in the system. In [68] an approximate dynamic programming approach is proposed to optimize for minimal cost of charging and power loss in a coordinated V2G charging scenario. One of the main concerns from users for V2G is the detrimental effect it may have on the battery life of their vehicle. In [6] a control scheme is proposed that provides the grid voltage and frequency support while also implementing a battery degradation model to minimize degradation issues and costs.

The research papers discussed above show how charging behavior can be changed to reduce negative effects of EVs on the electrical grid while still satisfying the demands of EV users. This is very important as the electrical grid can become a bottleneck in the journey to a carbon neutral society if not properly managed.

2.2 Charging Station Selection

When comparing charging of EVs to refueling of traditional gasoline-powered vehicles a significant difference is the time it takes to charge/refuel. This means an individual vehicle has a much larger effect on the waiting time at an EV charging station than at refueling stations. Because of this, it is important for EVs to have a good system for selecting a charging location.

The problem of charging station selection for EVs is very related to route planing algorithms for traditional vehicles, especially since the focus is on en-route charging. Optimal routing for vehicles is a field that has already been studied extensively. Simple systems use Dijkstra's Algorithm to compute the shortest path from source to destination. However, due to multiple other factors such as traffic, the shortest path is not always the quickest. Therefore, many algorithms combine the shortest path with real-time data on traffic flows. In [44] the authors formulate the shortest driving time problem as an integer programming model where alongside the distance between nodes in the network the current average speed on that edge is considered as an input. The model is then solved using a genetic algorithm and is shown to be accurate and performs well on systems with limited computational power and memory. Alongside real-time data, historical data can also be used as an input as shown in [71]. Here Ant Colony Optimization and a meta-heuristic approach are used to calculate the optimal path with the lowest travel cost.

When looking at algorithms specifically designed for charging station selection, the studies can generally be split into two groups: centralized and decentralized. Centralized algorithms revolve around a central entity that gathers information from all EVs looking for a charging station and tries to divide them in such a way that maximizes its objective function. Due to the nature of the centralized system it generally performs better than decentralized alternatives as shown in [12, 56, 70], as they have access to more information to make optimal decisions. However, practically there are many problems with centralized algorithms. From a user perspective there are concerns about privacy when having to share information related to current car state and charging preferences [15]. Manufacturers would also be required to all use the system, which could prove difficult with regards to agreements for funding as well as removing aspects in which manufacturers can differentiate themselves from competitors. Finally, as discussed in [56], a centralized system would scale poorly with increasing number of vehicles and the requirement of having full knowledge of all vehicle charging demands is unrealistic.

On the other hand decentralized solutions also come with their share of challenges. Due to limited available information the system can in most cases not reach an optimal solution. Decentralized approached to charging station selection are proposed in [16, 55, 56]. In [55] the authors try to achieve minimal waiting time by balancing the charging demand across all stations. This is done by requesting the minimum waiting time at the station closest to its current position. This station computes its waiting time and sends a similar request to the next station on the route to the vehicles destination. When this process reaches the final station on the route the messages are forwarded along the reverse route until the minimum waiting time of stations across the route reaches the vehicle. This approach achieves the goal of balancing loads across the charging stations. However, it is implemented from the perspective of the charging station, and would require all different operators of charging stations to work together be successful. The authors of [56] propose an algorithm that optimizes for cost saving for the user and load balancing for charging station providers. Vehicles evaluate the required time and energy required to visit all possible charging stations and select the one that minimizes charging cost, and make a booking. Meanwhile charging stations calculate their pricing based on congestion data found from the booking data. In this system vehicles are required to make a reservation at the charging station which introduces additional challenges such as vehicles not arriving on time affecting all future reservations, while with the system proposed in this thesis no blocking reserrvations are required. A mix of centralized and decentralized algorithms is proposed in [16]. Here each vehicle periodically receives probabilistic waiting times for each charging station from a central system. Using these waiting times and time-dependant stochastic data on traffic, vehicles derive a routing policy that minimizes the expected waiting time. From this policy expected arrival times can be computed which are fed back to the central system, which is then used to update the waiting time distributions that are fed back to the vehicles. This cycle repeats, and because the set of vehicles changes over time, the policies do not always converge. The proposed system shows promising results when compared to other selection algorithms. However, it still has a dependency on a central system to aggregate the arrival time distribution from all vehicles to calculate the expected waiting times. Whereas in this report a truly completely decentralized approach is presented.

2.3 Simulations & Models

For the development of new ITS it is important have realistic traffic scenarios, which are used to verify the correctness and performance of the system. Since developing and testing these applications with real vehicles is a costly, complex, and possible dangerous endeavor the development process utilizes traffic simulations. Traffic simulations are often divided into three categories [7]:

- *Macroscopic Simulators*: Simulate traffic flows from an aggregated point of view based on continuum traffic flow theory characterized by the variables: density, volume and speed.
- *Microscopic Simulators*: Simulate traffic flows based on the description of the motion of each individual vehicle in the traffic stream.
- *Mesoscopic Simulators*: An intermediate solution consisting of simplifications of microscopic elements such as vehicle dynamics, while still simulating individual vehicles.

Choosing the appropriate simulator for the application is a trade-off between factors such as accuracy of the simulation, computation time and freedom of control of the simulated entities.

The requirements for this project include simulation of the behavior of individual vehicles, as well as vehicle dynamics to accurately model the current state of the battery of an EV. Therefore, the focus will be on microscopic simulators. A popular, and open source, microscopic traffic simulator is Simulation for Urban Mobility (SUMO) [45]. SUMO is mainly maintained by developers of the Institute of Transportation Systems at the German Aerospace Center (DLR). Contributors to the project have done extensive research into various aspects of microscopic traffic simulation, such as car-following models [38, 39] and lane-change models [21, 37], in order for the simulation to be as realistic as possible. The SUMO simulator has been used to develop various ITS, for instance in [67] an optimization of the flow of vehicles through a network of signalized intersections is implemented. SUMO has also been used to simulate scenarios involving vehicle-to-vehicle communication, such as in [8]. Here SUMO is used in combination with the network simulator ns-2 [50] to evaluate the performance of routing protocols for vehicular ad-hoc networks (VANETs). In addition to simulation of traffic flows SUMO also support simulation of various vehicle dynamics. Most interesting in the scope of this thesis is the state of the battery in an EV. The energy model implemented in SUMO is described in [41] and is based on a sophisticated vehicle model that takes into account mechanic and electric vehicle parameters to calculate variations in depth of discharge between discrete time steps.

The final element that needs to be considered is the charging behavior and modelling of the battery, which is typically a lithium-ion battery for EVs. Recent developments in EV charging have been focused on fast charging to avoid

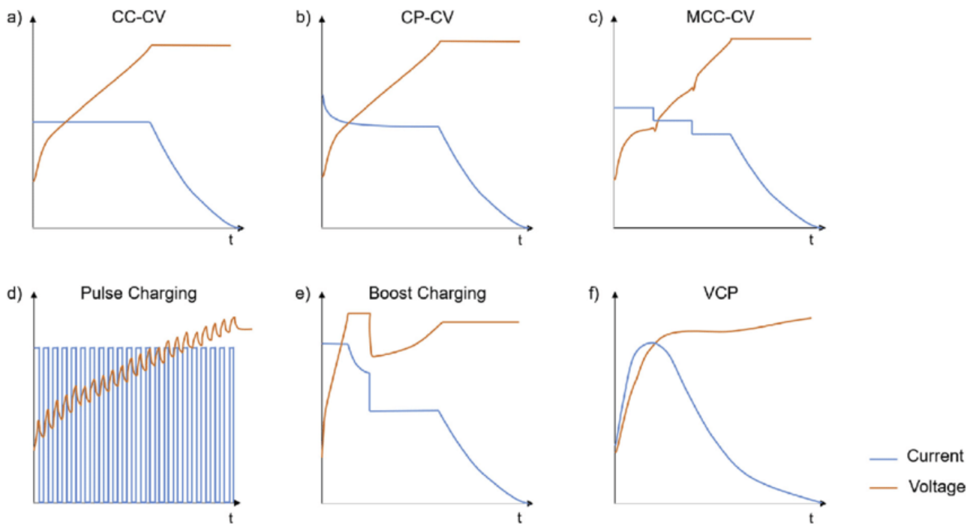

Figure 2.2: **Representation of common types of charging protocols [62]. a) constant current - constant voltage, b) constant power - constant voltage, c) multistage constant current - constant voltage, d) pulse charging, e) boost charging, f) variable current profile**

minimize the long charging times that are often associated with using EVs. However, fast charging has been shown to have negative effects on the batteries, accelerating degradation of capacity and power capabilities [62]. In order to minimize charging times as well as degradation of the battery performance different charging protocols have been researched, some common types of these are shown in Figure 2.2. Battery models are an important input to the battery management system (BMS) of an EV. The BMS is a critical system that interfaces directly the the vehicle's battery, its responsibilities include reporting state-of-charge (SoC) and state-of-health (SoH) of the battery as well as detecting faults. The SoC of the battery plays an important role in selection of a charging station as it determines which charging stations can be reached, charging time as well as when to charge. Measuring SoC is not a trivial task and there are different implementations, depending on the chosen battery model, each with its pros and cons [5]. All the aforementioned components together can be used to create to an accurate model of an EV battery and its characteristics under charging and driving conditions.

Chapter 3

System Architecture

The exploration into the current state-of-the-art in the previous section has shown that a lot of work has already been done in the optimization of both the EV charging process itself as well as the selection of charging stations for en-route charging. The validation of these proposed systems can be done with existing simulators, which are available for the fields of large-scale traffic simulation, V2X communication, and battery characteristics.

The thesis project described in this report looks to explore how V2X can be used to improve en-route charging for users of Porsche EVs. The metric the improvements will be measured with is the total journey time from origin to destination. In the case of a charging event happening the journey time consists of three sub-components: driving time, charging time, and waiting time. Of these three components the driving and charging times are predominantly determined by the distance between origin and destination and the initial SoC of the battery. This leaves the waiting time as the component where the most difference can be made. Since the waiting time depends on the behavior of other vehicles in the area, V2X communication is perfectly suited to help minimize this.

In an example scenario, charging station i might be more on the route of the ego vehicle than station j. However, if station i is busier, then the extra waiting time might be more than the added travel time to visit station j. In this case visiting charging station j will result in a lower total journey time. In order for vehicles to be able to make a decision on whether it is beneficial for them to take a bigger detour to achieve a lower total journey time they need to be able to make an accurate estimation of the waiting times at each charging station. This is where V2X comes into play. The EVCSN message as defined by ETSI contains the current minimum waiting time for each charging spot at a charging station, which gives a rough idea of the vehicle distribution across charging stations. However, with many vehicles looking for a charging station and receiving the same information this might not give an accurate description of what waiting time would look like when actually arriving at the charging station. In order to bridge this gap an extension is proposed to the EVCSN message to also include information about vehicles currently travelling towards the station for charging. This information is obtained by vehicles informing charging stations of their charging intentions using the EV-RSR procedure.

In order to evaluate the performance of different charging station selection strategies under the current EVCSN definition, as well as the proposed extension to include information about the charging intention of other vehicles, a simulation is necessary that is capable of simulating large numbers of vehicles as well as V2X communication. Currently Porsche Engineering has two possible ways of evaluating V2X applications. The first is using test vehicles available for the development of Advanced Driver Assistance Systems (ADAS). These are known as JUPITER vehicles (Joint User Personalized Integrated Testing and Engineering Resource) and are equipped with additional sensors such as LiDAR and cameras. Both the standard and additional sensors are connected on a ROS network, and the vehicle has recently also been equipped with V2X technology allowing for testing of V2X applications in real environments. The second option is to use the internally developed simulation environment known as PEVATeC SimFramework (Porsche Engineering Virtual ADAS Testing Center Simulation Framework), which is based on the open-source CARLA simulator [17]. This simulation framework allows for detailed simulation of various traffic scenarios with small numbers of vehicles, and features a digital twin of the JUPITER vehicle that has the same sensors as the real vehicle allowing for accurate simulations. Unfortunately, neither of these solutions is suitable for the evaluation of a scenario with a large number of vehicles. Therefore, a new simulation framework will have to be set up that will be used to evaluate the proposed charging station selection strategies. This will be used to asses the effect the different strategies have on the total journey time of vehicles in the scenario. Additionally, the JUPITER test vehicle will be used to give a demonstration of how the system would be integrated, as well as showing it is viable for implementation in real vehicles.

In the remainder of this section the architecture of the developed systems is discussed. The work done during this project can be split into three components, which will be discussed separately. The first component is the simulation framework that combines simulation of large numbers of vehicles, and V2X communication. Secondly, the different strategies for the selection of charging stations will be discussed. Finally, the architecture of the integration of the simulated environment with the JUPITER test vehicle will be explained.

3.1 Simulation Framework

In order to get a good idea of the performance of the selection strategies a test has to span a longer period of time, allowing multiple waves of vehicles to pass the area containing the charging stations. As such a test would require a large number of vehicles, people and infrastructure a physical test is not feasible and thus a simulation is used. Since en-route charging occurs in cases where the vehicle is not able to reach the destination with the remaining battery charge the simulated environment should ideally be sufficiently large to simulate such longer journeys. Additionally the simulation should support the transmission of V2X messages between entities in the simulation. For the simulations required to evaluate the system proposed in this thesis only communication between vehicles and charging stations is required. However, since it is desirable for Porsche Engineering to have a simulation environment capable of handling large numbers of vehicles to be able to be used for other projects in the future, the V2X communication should be extendable to all entities in the simulation.

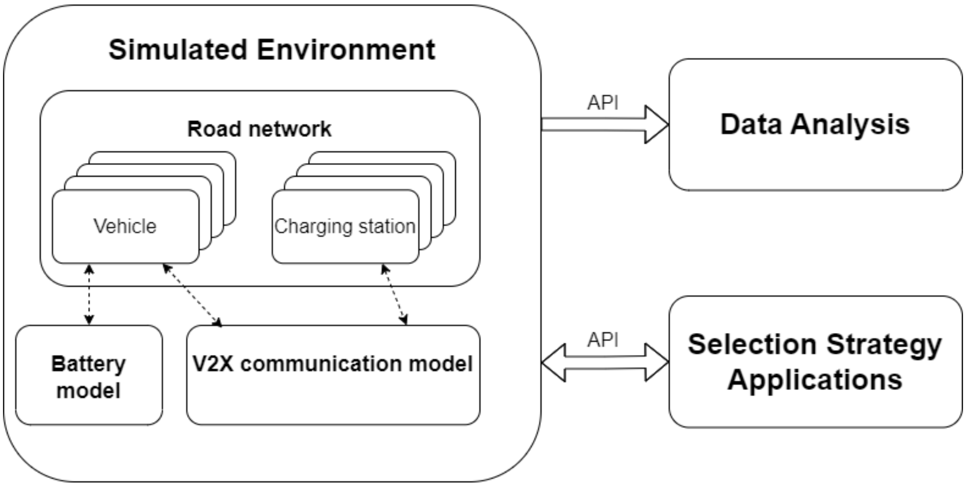

Figure 3.1: **Overview of the simulation environment, and the required interfaces for evaluating the proposed V2X based selection strategies.**

Vehicles in the simulation should follow a set a route from origin to destination such that results can be reproduced and predictions on performance can be made. The transmission of V2X message should ideally be realistic, such that analysis can be done on the effects of factors such as vehicle density and coverage of V2X infrastructure. However, for the purpose of verifying the performance of the charging station selection strategies a simple model that only transmits data from one entity to another without accurately simulation the message transmission would suffice. Finally, since the problem concerns EVs, and an important factor in the decision of which charging station to use and whether to charge is the SoC of the vehicle, there should be a component of the simulation that models the battery of a vehicle and updates the SoC as it drives through the network.

The research into the current state-of-the-art covered in Section 2 showed that there are simulators and models available for traffic simulation, V2X communication and the dynamics of EV batteries and charging meeting the requirements described above. This means it is not necessary to build a simulation from scratch, but only to link these simulators and models together to create a complete simulation. A high-level overview of the different components required to be able to run simulations that can evaluate the performance of the proposed strategies is shown in 3.1.

The linking of different simulators and models each describing a small portion of the complete problem is called co-simulation, and in the field of connected mobility there exists co-simulation tools that combine traffic simulators, network simulators, and provide an API to create custom applications linked to entities in the simulation. The co-simulation tool selected for this project is Eclipse MOSAIC [58] as it supports the use of SUMO for traffic simulation, and NS-3 for network simulation. This combination of tools was recommended by the function owner of the V2X project at Porsche Engineering as he had experience using these tools in previous projects and could provide technical support, as well as knowing this setup would be able to meet the requirements that were set.

In order for the simulation to give results that would translate best to the real-world, attempts were made to make the simulation as realistic as possible. This meant modelling the road network in the simulation based on existing infrastructure. This also has the advantage that it would be possible to gather data on real traffic conditions on the area to use for vehicle routing. As the goal was be to integrate the simulation with the JUPITER test vehicle, the chosen area to model the road network on was the area around the Porsche Engineering facility where the JUPITER vehicle was located. This way locations of charging stations in the simulation would be in relatively close proximity to the GPS location of the JUPITER vehicle, which is available as a ROS topic. For this area no publicly available traffic data was found and therefor no realistic traffic conditions could be created in the simulation. It is possible that this data is available at local authorities or could be determined by other means, but it was decided that random routes through the road network would give a good enough impression of the performance of the system.

The SUMO traffic simulator has a model implemented for an electric vehicle. This model takes vehicle characteristics such as vehicle mass, surface area, and power to compute changes in battery SoC while driving. With this vehicle model also comes an implementation of charging infrastructure. The charging infrastructure implemented in SUMO was created with the purpose of evaluating inductive charging solutions. Therefore, the charging stations that are available are limited to placement on roads and used to charge vehicles in transit. This does not match the requirements for this project as traditional charging stations where vehicles are parked for the duration of charging are considered. For this reason the charging of vehicles is instead handled by external applications using the API to modify the SoC of vehicles each timestep. This means some extra work has to be done to get the charging working, but would give more flexibility afterwards to modify the exact charging behavior.

3.2 Selection Strategies

The main objective of the thesis project is to develop and evaluate charging station selection strategies based on V2X communication between vehicles and charging stations. Before getting into detail on the strategies that have been developed it is important to establish some baseline strategies to make comparisons against. For the evaluations, which will be discussed in detail in Chapter 5, two baselines were chosen to compare the V2X based strategies against. These are a naive scenario in which vehicles only have access to the location of charging stations, and an optimal scenario where all of the charging stations have unlimited capacity. These two options should give an upper and lower bound for performance within the scenario used for evaluation. The charging station selection strategy for both of these scenarios does not take into account the waiting time at charging stations, as no V2X is used and therefore this information is not known. Instead, the only information that is used to make a decision is the location of the charging station and the amount of time adding that location as a stop en-route to the destination would add to the total journey time.

The remaining strategies all use information gathered from V2X communication, specifically from the EVCSN message. The EVCSN message as defined by ETSI contains the current minimum waiting time for a charging spot at a charging station. This information can be used by vehicles to assess the traffic at each of the charging stations and distinguish between charging stations that are busy and those that are not. However, this only covers the current state. If the ego vehicle, which is a term used to describe the main vehicle in a scenario, is currently far away from the charging station the current state might not accurately represent the waiting time when the ego vehicle arrives at the charging station. In order to bridge this gap between the current waiting time and an accurate estimation of the waiting time when the ego vehicle reaches the charging station two solutions have been considered.

- The first solution does not require any changes to the V2X message definitions and simply consists of periodically re-evaluating the data received from the EVCSN messages. This way as the state of the charging stations change due to vehicles arriving for charging, the ego vehicle could get rerouted to a different charging station as that would now result in a shorter total journey time. These two methods of routing to the charging stations will be referred to as one-shot (OS.) and continuous (CONT.) navigation.
- The second solution is extending the definition of the EVCSN message to include information about the intentions of other vehicles. With the use of the EV-RSR procedures, vehicles can share their charging intentions with a charging station. These procedures are intended for making blocking reservations in a specific timeslot. However, in this case it will be used for the purpose of sharing intentions. Changes required to the EV-RSR message definitions to make the difference between intention sharing and reservations are not discussed in this documents as in the context of this project the two use-cases were not use simultaneously. Instead, intention sharing was assumed to be the only use of EV-RSR messages. Scenarios with the updated EVCSN message will be referred to as extended (EXT.).

Using the different solutions discussed above a total of 4 different permutations are possible for V2X based strategies (V2X OS., V2X OS. EXT., V2X CONT., V2X CONT. EXT.). These strategies will be compared against the optimal and naive scenarios to evaluate their effectiveness in reducing the total journey time for en-route charging of EVs.

When extending the EVCSN two main concerns where kept in mind. The first being adding the least amount of data necessary. This is done in order to keep the bandwidth usage of the EVCSN as low as possible. Secondly, the data that is added to the EVCSN message should be anonymous. The EV-RSR procedures, designed for making charging reservations, are unicast messages such that malicious agents cannot easily get information about a vehicles behavior. When information about intentions of other vehicles are shared through the EVCSN message this same concept should apply. In order for the ego vehicle to make an accurate estimation about waiting times in the future it should know three things about the state of the charging station: the current minimum waiting time, the times at which other vehicles arrive at the station, and the time these other vehicles will spent charging at the station. The first is already included in the EVCSN message definition, and the other data does not contain any information with which the other vehicles could easily be identified. In addition to the

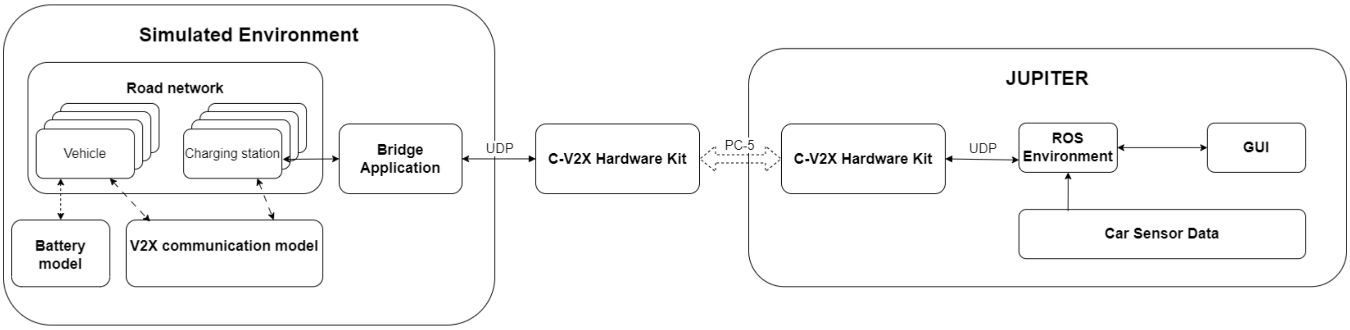

Figure 3.2: **Overview of the integration of the JUPITER test vehicle with the simulation environment using a bridge application to connect to the V2X hardware.**

required data being anonymous, adding two timestamps for each vehicle does not have a significant effect on the bandwidth usage of the message. Therefore satisfying both of the goals for the extension.

3.3 Test Vehicle Integration

An important aspect of the thesis project was to show that the designed system would work with real C-V2X hardware and a test vehicle. Doing this shows that the system is viable for real implementation and also gives a good impression to the management that will eventually decide which V2X projects to pursue for implementation in production vehicles. For testing and demonstration of V2X projects the JUPITER test vehicle, originally developed for testing of ADAS projects, has been equipped with a C-V2X hardware kit connected to the onboard development PC. The hardware kit runs a light weight Linux operating system and is shipped with a basic application to send and receive V2X messages. To connect the V2X hardware to the vehicle, in order to fill V2X messages with up-to-date data and extract data from incoming messages, UDP sockets are used.

For a successful demonstration the test vehicle should be able to receive information about the charging stations in the simulation and their states through EVCSN messages. This data should be presented using a graphical user interface (GUI) on the dashboard display in the vehicle. The data displayed should show the user the available charging stations, the waiting times at those charging stations, and provide a recommendation to the user for which charging station to visit. The algorithm used for the recommendation should have the same functionality as the vehicles in the simulation. A high-level overview of the system is shown in Figure 3.2.

To perform this demonstration, a number of systems and applications will have to be created. First of all, the simulation should be extended to allow for communication with the V2X hardware. This communication happens over UDP. The charging station applications in the simulations should, in addition to transmitting an EVCSN message in the simulation, transmit the data that is included in the EVCSN on the UDP connection for the V2X hardware to assemble an EVCSN message of its own. The V2X hardware on the simulation side will act as an RSU that gathers information from multiple charging stations and aggregates it into a single EVCSN message. In addition to sending the EVCSN it will also receive EV-RSR messages coming from the JUPITER vehicle. These are again communicated to the simulation through the UDP link, where they will be processed by the charging station applications. In contrast to inside the simulation, when using the real hardware no guarantees can be made about the successful transmission of the V2X messages. Therefore, the EV-RSR messages that are used will always be acknowledged by the receiving entity as described in the EV-RSR specification [23].

The application running on the V2X hardware will have to be extended to support sending and receiving of EVCSN and EV-RSR messages. Definitions for custom messages can be provided using the ASN.1 language. ETSI hosts a repository containing all their proposed messages in the ASN.1 language [24].

Finally, on the JUPITER development PC a ROS node is used that collects the necessary data from the car sensors. This data is collected from the car sensors and published on ROS topics by an existing application. The car data that is required are the current GPS location and the SoC of the battery to perform routing and charging time calculations, respectively. This data is combined with the incoming data from the V2X hardware, containing the charging station information, that is connected to the PC and shown to the user on the vehicle display with a basic GUI.

Chapter 4

System Overview

This chapter will provide details on how the system outlined in Chapter 3 is implemented. First, the relevant V2X messages are discussed as well as how they were modified to fit the purpose of this project. Afterwards, the implementation of the simulation environment is shown, as well as how the different strategies are implemented. Finally, the integration of the JUPITER test vehicle and the simulation environment is detailed.

4.1 V2X Messages

In order to achieve the goal of reducing the journey time when charging, en-route vehicles need to be able to know which charging station will add the least time to their journey. This is a combination of added travel time, which can be found from static map data or supplemented with dynamic traffic data, and the time spent at the charging station itself. The time spent at the charging station again consists of two components: time spent charging and time spent waiting. In this case the focus will be on minimizing the time spent waiting as charging time is mostly dependant on static characteristics of the charging station, whereas waiting time is dependant on the behavior of other vehicles. One way of gathering information about the current state of the charging station is through V2X communication between the vehicle and the charging station. For this particular use case ETSI has defined two sets of V2X message standards: the EV charging spot notification (EVCSN) and the EV recharging spot reservation (EV-RSR) procedure.

4.1.1 EVCSN

The goal of the EVCSN message is to notify drivers about the existence and characteristics of one or more charging spots in the area [22]. This includes both static and dynamic characteristics. The contents of the message as proposed by ETSI in ASN.1 format can be found in Appendix A. There are two main scenarios for the sending of EVCSN messages. The first is that a V2X capable roadside unit (RSU) is associated with a single charging station. Here the charging station and RSU are directly connected and the message broadcasted by the RSU contains the data of only that charging station and any following

```
ItsChargingStationData ::= SEQUENCE {
    ...,
    futureArrivals      FutureArrivals,
    ...
}

FutureArrivals ::= SEQUENCE (SIZE(0..10)) OF FutureArrival

FutureArrival ::= SEQUENCE {
    arrivalTime     TimestampIts,
    chargingTime    ChargingTime
}

-- time in msec (max of 24 hours)
ChargingTime ::= INTEGER (0..86400000)
```

Listing 4.1: **Modifications made to the EVCSN message, in the ASN.1 language**

exchanges regarding reservation can be handled directly by the RSU. In the second scenario the RSU is centrally positioned and receives information about several charging stations in its surrounding from some central station that the charging stations are connected to. Here the message broadcasted by the RSU contains data from multiple charging stations and reservations are forwarded by the RSU to the charging station.

In the current specification provided by ETSI each charging station can have multiple charging spots, with possible different charging equipment and/or charging power, which can have multiple parking spots for EVs to use for charging. For each of these parking spots the EVCSN contains a minimum waiting time. This waiting time is the sum of the remaining time to charge the vehicle currently being serviced and the time required to charge any waiting vehicles. Communication, through the charging cable, between the charging equipment and the BMS of the charging vehicle can be used to determine the remaining charging time. Additionally, charging stations could also utilize camera systems or induction loops to estimate the number of vehicles that are currently waiting in the case that all charging spots are in use. The exact implementation of such a system is out of scope for this project, but the assumption is that the charging station is able to accurately report the minimum waiting time for its charging spots.

On top of the minimum waiting time, based on vehicles currently at the charging station, the system proposed here extends the EVCSN message to also include information about vehicles that are currently on their way to charge at the charging station. This information is tracked by the charging station based on reservation and cancellation messages received from vehicles. The relevant changes made to the message content are shown in Listing 4.1.

Future arrivals are tracked for the entire charging station as opposed to for each charging spot in the station. The reason for this is that future arrivals are not reservations, but vehicles sharing their intentions with others to allow for more informed decisions to be made. Charging stations use a first in first out queueing system, so a specific charging spot may no longer be available because of other vehicles arriving that do not participate in sharing their intentions, whereas another spot could still be free. Combining this with the fact that in the evaluated scenarios all charging spots have equal characteristics there is no added benefit for selected a specific charging spot within the charging station.

4.1.2 EV-RSR

The EV-RSR messages are covered in an ETSI technical specification on communication system for planning and reservation of EV charging [23]. This document describes the a reservation process along with the messages that would be required to implement it. The reservation process extends from the start of the journey all the way to arrival at the charging station. The actual reservation procedure starts once a candidate reservation spot has been selected. This selection is done based on information provided to the vehicle through the EVCSN message discussed in Section 4.1.1. The first step in making a reservation is to identify the reservation server that handles the requests, this can be found in the contents of a received EVCSN in the case that one notification message includes multiple charging stations or the requests can be sent directly to the RSU corresponding to the charging station if they are one-to-one linked.

The first step in the procedure is referred to as pre-reservation. The vehicle initiates the procedure by sending a pre-reservation request for a specific charging spot. The server responds with the availability of the selected spot, putting it in a pre-reserved state for a set time if its available. The next step is for the vehicle to send a reservation request. This contains a pre-reservation ID received from the server as well as a payment method and energy requirements for the charging session. If the charging station can service the requested charging session the vehicle receives a response message with response code OK, or an error code otherwise. If after confirmation of the reservation the vehicle wishes to cancel or update the reservation additional procedures are defined in the technical specification for these purposes. The data contained in these message as proposed by ETSI can be found in Appendix B.

In the scope of this project the reservation procedure as described in the technical specification by ETSI is not entirely appropriate. Instead of making a reservation, which comes with additional challenges such as enforcing that the spot is not occupied when a vehicle with a reservation arrives and that vehicles leave after the end of the reservation period, in this case vehicles only need to share their intention to charge at the station alongside arrival time and charging duration. Therefor, the pre-reservation procedure is omitted and only the reservation, update and cancellation procedures are used. In order to make a distinction between a message that shares a charging intention and a message for a reservation request a flag is added to the reservation request message.

4.2 Simulation

In order to create a system that can achieve the goals outlined in Section 1.1 a simulation environment is needed that covers simulation of vehicles travelling through a road network, messaging between these vehicles, and a way to manipulate the behavior of the vehicles based on their current state. This requires

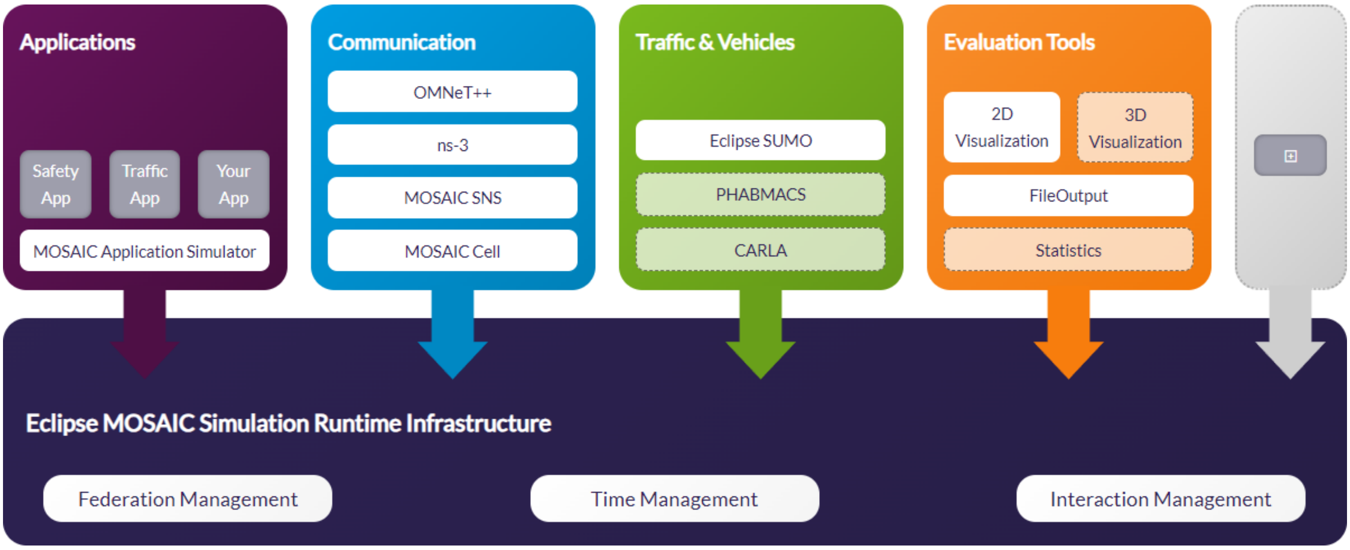

Figure 4.1: **Overview of Eclipse MOSAIC structure, showing the different simulation domains, the available simulators and how they are centrally managed by Eclipse MOSAIC [20].**

the use of co-simulation. In co-simulation different subsystems are handled by different simulators, allowing the use of detailed and specific simulators for each subsystem instead of requiring one simulator to cover all of them. The chosen co-simulation tool for this project is Eclipse MOSAIC [58].

4.2.1 Eclipse MOSAIC

The core of the simulation environment is provided by the Eclipse MOSAIC simulation framework. The architecture of Eclipse MOSAIC is based on the high-level architecture (HLA) standard [35], the main components of which are the runtime infrastructure (RTI), federates and ambassadors. A federate is a wrapper object for an individual simulator and allows for a standardized interface between the simulator and the RTI. The communication between the simulator and RTI is handles by ambassadors. Typically a federate includes two ambassadors: a federate ambassador, which allows the RTI to control the simulation and handles interactions from other federates, and a RTI ambassador, which allows the federate to access services provided by the RTI such as sending interactions to other federates and requesting the current simulation time. Finally the RTI is responsible for the management of all federates, this includes starting and stopping federates, managing interactions between different simulation units to exchange data and time management to ensure synchronization between federates. An overview of the MOSAIC architecture, and the supported simulators, can be seen in Figure 4.1.

In order to use MOSAIC to run simulations a scenario has to be created. This scenario consists of data and configurations for the RTI, and each of the federates. The main components that make up a scenario are the scenario database, traffic simulator- and mapping configurations. The scenario database is an SQLite database containing information on road geometry, connections between nodes in the network, buildings and routes. The database is accessible by federates through the RTI ambassador and can for instance be used for route calculation in the application simulator or in sophisticated communication models. Configuration for the traffic simulator depends on which simulator is used. In the case of SUMO the network files have to be provided as well as configuration for simulation step size, and any additional parameters to be passed to SUMO on start. The mapping configuration handles the definitions and creation of simulation entities. MOSAIC supports the following entities under the open-source license: vehicles, road-side units (RSUs), traffic lights, servers and traffic management centers. Besides the open-source license MOSAIC also has a commercial version which includes a charging station entity as well as a battery simulator and some additional tools for analysing simulation results (for this thesis the open-source version was used). The entity types that are relevant for this project are vehicles, RSUs and servers. The mapping file allows defining prototypes for the different types of entities and specifying their parameters as well as applications that should be mapped to these entities (applications are discussed in more detail in Section 4.2.4). These prototypes are then reused in the later sections of the mapping file resulting in an easier to manage, and shorter configuration file. In addition to defining prototypes for entities, the mapping file also specifies instances of entities. Definition of instances are split into sections depending on the type of the entity. Vehicle instances are defined by a spawner that can be configured using a route, distribution of used vehicle prototypes, spawning frequency and a maximum number of spawned vehicles. RSUs require GPS coordinates and a prototype to be provided, and servers only require a prototype.

4.2.2 SUMO

SUMO has been selected as the traffic simulator for the project because of its integration with MOSAIC, and experience within the company. SUMO is a microscopic traffic simulator that allows for simulating individual vehicles in a large road network. An example of a road network in SUMO with simulated vehicles as seen in the SUMO GUI is shown in Figure 4.2. In order to create a simulation scenario multiple elements are needed, which are discussed in the following sections.

Road network

A basic SUMO network consists of nodes (junctions), and edges that connect the nodes. There are multiple ways to create networks in SUMO. The first option is to manually create networks using the provided netedit tool. Netedit is a visual network editor in which all aspects of network can be created from scratch. The second option is to use the netgenerate tool, which allows for generation of grid, spider, and random networks based on a multitude of parameters. Finally networks can be imported from other sources using netconvert. Netconvert can convert road networks from other popular traffic simulators to the SUMO format as well as creating a network based on real world data from the free to use OpenStreetMap (OSM) [51].

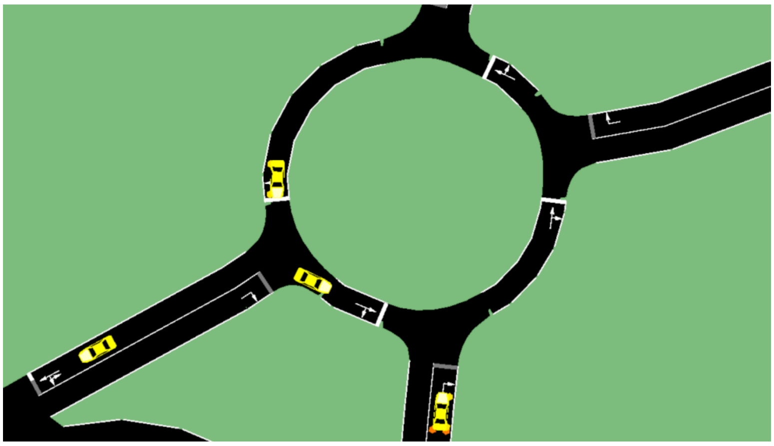

Figure 4.2: **Portion of a SUMO road network showing vehicles navigating a roundabout.**

Additional Infrastructure

Aside from a road network a detailed simulation required additional infrastructure. SUMO has support for a broad range of additional infrastructure elements such as induction loop detectors, bus stops, parking areas and charging stations. Of most interest in the scope of this project are the charging stations. The implementation of charging stations in SUMO was based on the pre-existing bus stops. Bus stops are defined as a part of a lane where vehicles can stop for a pre-determined time. Due to this implementation it is only possible to place charging station on road segments, which has the unwanted effect of blocking traffic on the lane. Therefore, instead of using the provided charging station elements, parking areas will be used. These can be added as stops on the route of a vehicle. Then, in the application attached to the vehicle a stop at a parking area will be detected, which triggers the start of the charging process.

Traffic

Vehicles are key to any traffic simulation, and these vehicles require a set of edges that defines the path they follow. These paths can be defined in two possible ways. The first is as a *trip*, which contains a starting edge, destination edge and departure time. The second option is a *route*, which instead of only containing the starting and ending edge contain all edges the vehicle passes.

SUMO offers multiple ways to create trips and routes. These can be based on real-life data made available by traffic authorities such as origin-destination matrices [52] or induction loop data, which can be converted to trips using tools provided by SUMO. Alternatively trips or routes can be generated by hand either directly in the XML files or using netedit. Finally, SUMO provides a python script for generation of random trips through a network along with many parameters to tune the generation to fit the purpose of the simulation.

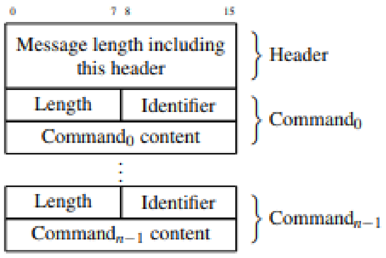

Figure 4.3: **TraCI message format [65]**

API

In order to manipulate the behavior of the vehicles in the simulation, based on the data they receive through V2X, it is necessary for SUMO to expose an API that makes this possible. To evaluate the performance of applications based on VANETs researchers have implemented the Traffic Control Interface (TraCI) to make it possible to couple a traffic simulator to a network simulator to perform these evaluations [65]. TraCI is also what is used by MOSAIC to control the SUMO environment for example by spawning vehicles.

TraCI uses a client/server architecture with TCP connections. SUMO acts as the server and starts listening for commands after starting, other simulators or applications can then connect to the TraCI server to send messages. TraCI messages consists of a small header followed by a variable number of commands as shown in Figure 4.3. Each TraCI request sent by a client generates on or more responses from the server after the commands in the request have been processed.

4.2.3 Network Simulation

The network simulator is responsible for modelling the transmission of messages between entities in the simulation. MOSAIC already comes with support for multiple network simulators and using the extendable federate structure allows users to integrate any other simulators as well. The preferred network simulator to use from the side of Porsche Engineering was ns-3. This simulator is commonly used in research, however does not natively support V2X simulation. A simulator based on ns-3 has been created specifically for C-V2X simulation in [19]. The goal was then to replace the standard version of ns-3 that was used by MOSAIC with the modified simulator. However, the main priority of the project was to verify the workings of the system as a whole, which does not require a very detailed network simulator. Therefore, the integration of ns-3 was added as a low priority task, which in the end could unfortunately not be completed. Instead a built-in simulator created by the developers of MOSAIC was used, called the simple network simulator (SNS). The simulator aims to provide a simple and performant implementation of V2X transmission, which fits the requirements of the project.

For the proposed system two different message types are required, the EVCSN and EV-RSR both of which are discussed in Section 4.1. A difference between these two messages is in the targeted entities. The EVCSN should be a broadcast message that is ideally received by all vehicles in the area, whereas the EV-

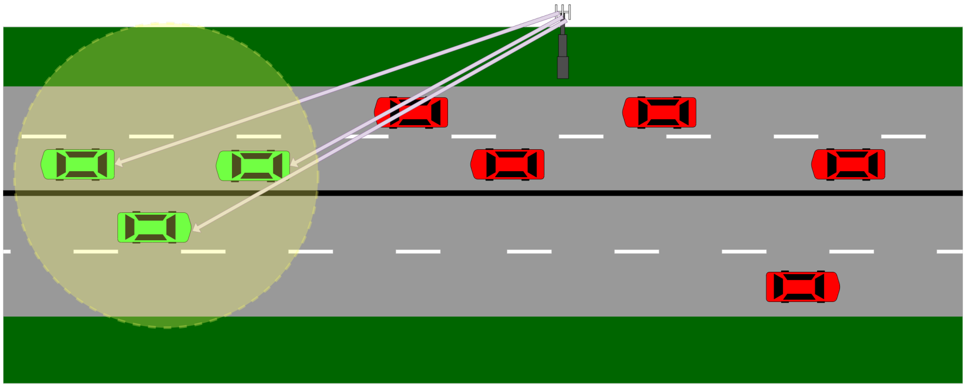

Figure 4.4: **Example of geobroadcast, RSU sending data to all entities in the highlighted area [49].**

RSR is directed towards a specific entity (either vehicle or charging station). For the sake of simplicity both have been implemented using the geobroadcast functionality of SNS, which is shown in Figure 4.4, with the difference that the EVCSN target area is the entirety of the road network and the EV-RSR is targeted at the location of the entity to improve performance.

4.2.4 Applications

Applications are the part of the MOSAIC environment where the different simulators come together and the behavior of the different entities in the simulations is defined. Through the mapping functionality it is possible to attach applications to all the types of entities that can be created in a simulation scenario. For the system discussed in this thesis two main applications types are required. These are the charging stations and the vehicles. Finally there is a data collection server that is built to assist in analysing the results of the simulation.

The applications for a MOSAIC scenario are programmed in Java and attached to entities in the simulation by the RTI after startup based on the configuration provided in the mapping file. Each instance of a application is a separate instance of a class. In order to implement the required behavior correctly there needs to be some communication between these instances. This can be done in two different ways. The first is using V2X messages sent by one application and through transmission using the network simulator can be received by the applications attached to entities in the targeted area. The second method is through interactions. There are different types of interactions supported by MOSAIC and they are for example used to notify a vehicle application of changes to the vehicle in the traffic simulator that it is mapped to. Interactions are used for behavior that in a real-life situation would be handled by hardware, cables or other systems whose implementation are out of scope this project.

Charging Stations

The behavior of charging stations can be summarized by two main features. These are the tracking of the current state of the charging station and sending the EVCSN messages. The EVCSN messages are sent periodically by calling a function of the MOSAIC API that sends an interaction to the network simulator to transmit the message. The target area of the message is the entire simulated area. Normally the range would be more restricted if using a cellular-based broadcast or the PC5 sidelink and would require a network of RSUs to reach coverage of the entire area. But for the purpose of this project all vehicles in the simulation would be able to receive the message. The state of the charging station consists of the number of vehicles that are currently there to charge and the vehicles that have used an EV-RSR message to indicate they are travelling to the station to charge.

Incoming and outgoing vehicles are tracked using interactions sent by the vehicle whenever they arrive at or leave the station. This way all vehicles at the charging station are known, both those that are currently charging and those waiting for a charging spot to become available. In reality it would be a bit more difficult to keep track of both these things. Tracking vehicles currently charging is relatively trivial as that could be derived from the charging spots and communication happening through the charging cable. The difficult part is knowing the number of vehicles waiting for a charging spot. Some possibilities for this implementation are induction loops and camera systems monitoring the charging station area, but the specifics of such a system are out of scope. Using the number of vehicles currently at the charging station the current minimum waiting time can be computed, which is a key component of the EVCSN message content.

Future arrivals tracking is done based on received V2X messages. The message that are used for this are the EV-RSR messages. As in simulation we can guarantee that messages always reach the targeted entity the response messages defined in the EV-RSR specification as discussed in Section 4.1.2 are left out as they would just add unnecessary complexity. The two messages from the EV-RSR specification that are used are the reservation request and the cancellation, which add or remove a vehicle from the list of future arrivals respectively.

Vehicles

All EVs in the simulation will have an application mapped to it to control its behavior with regards to charging. On initialisation each vehicle will set its initial state of charge (SoC) and a threshold SoC value at which point it will want to charge. The initial SoC is sampled from a beta distribution with parameters $\alpha = 3.09515$ and $\beta = 2.9839$. This approach is taken from [33] using a max range of 410km based on the Porsche Taycan and a daily-used range of 9.5km based on the average length of the routes used during the simulation. The threshold SoC is based on [18], which shows an almost gaussian distribution centered around 50% for the starting SoC of charging events. Both of these values are clamped to prevent events such as vehicles spawning with an empty battery or constantly charging if their threshold is set at 99%.

As the vehicle starts driving along its route it constantly monitors its SoC. Once it drops below the threshold value it starts looking for suitable charging stations and will try to pick the one that adds the least time to the journey. In order to compare the effects of using V2X, and the modification made to the EVCSN to include future arrivals alongside the current waiting times there are different selection strategies a vehicle can use. There are five different strategies implemented in the simulation based on different combinations of the following

Listing 4.2: **Future waiting time prediction routine**

```
function PREDICT_FUTURE_WAIT_TIME(curWaitTimes, futureArrivals)
    initialise futureWaitTimes to 0 for each cs
    for each cs_i do
        waitTimes ← curWaitTimes_i
        t_last ← curTime
        sort futureArrivals_i by arrivalTime
        for each futureArrival f in futureArrivals_i do
            if arrivalTime_f > myArrivalTime
                break
            dt ← arrivalTime_f − t_last
            t_last ← arrivalTime_f
            for time t in waitTimes
                t ← max(0, t−dt)
            sort waitTimes in ascending order
            waitTimes[0] += chargingTime_f
        dt ← myArrivalTime − t_last
        for time t in waitTimes
            t ← max(0, t−dt)
        sort waitTimes in ascending order
        futureWaitTimes_i ← waitTimes[0]
    return futureWaitTimes
```

properties:

1. V2X capabilities
2. Routing method (only for V2X capable vehicles)
3. Use of EV-RSR (only for V2X capable vehicles)

A V2X capable vehicle will periodically receive EVCSN messages from charging stations while on its journey, and use the information it has available from these messages to make a decisions on which charging station is estimated to add the least travel time. Vehicles that do not have V2X only have access to static information about the charging stations in the area, being their location. Using these locations it queries the TraCI API to compute routes from its current location to the charging station and from the charging station to the final destination. Using the route from current location to the charging station the vehicle first evaluates if it possible to reach the charging station with the energy remaining in the battery. The required energy to drive a certain route is based on the average speed driven on the route derived from the driving time and driving distance that are returned by the routing API. From the reachable stations the one is selected for which the combination of the two calculated routes has the least travel time.

Vehicles that do have V2X will, instead of only looking at the travel time for the journey to and from the charging station, also take the waiting time at the charging station into account. Picking the charging station that has the lowest sum of driving time and waiting time. For the estimation of the waiting time there are two options. When EV-RSR messages are not used in the system the vehicle uses the current waiting time for each station as found in the EVCSN message data. In the case that EV-RSR is being used the EVCSN will also contain a list of future arrivals. Using this data the vehicle tries to predict the waiting time at the charging station at the time it would arrive using the current waiting times as the initial state. In doing this it assumes that the future arrival list is accurate and no other vehicles will arrive at the station that has not already shared its intention to go there. The algorithm for this prediction is shown in Listing 4.2.

The final variation between vehicles that is possible is in the routing method. The difference here is between one-shot and continuous navigation. When using one-shot navigation a charging selection will be selected once and no more changes are made to the route after that. Whereas with continuous navigation, new information is constantly evaluated whilst the vehicle is driving to the charging station. This new information could lead to the vehicle deciding that a different station is now considered the best choice. If this happens the route is adjusted and in the case that EV-RSR is used a cancellation message is sent to the old charging station before sending a reservation to the new station.

Server

In order to analyse some aspects of the simulation it is required to get some data gathered throughout the whole runtime of the simulation. In MOSAIC each instance of an application creates its own log folder and log files, to which its also possible to add custom log entries. However, since vehicles enter and leave the simulation constantly there is no one single vehicle that can log for the entire duration of the simulation. Charging stations suffer from the problem that there are multiple instances of them and thus would create a lot of duplicate data, which due to the format of the logger causes a significant increase of log size. Additionally, it is important to know whether a vehicle was destroyed because it reached the end of the simulation, or because the simulation ended. This is because during evaluation, vehicles that still in transit or in the middle of a charging station should be filtered out of the results. Data related to the journey of the vehicle such as its total journey time can also not be logged by vehicles that are destroyed by the simulation ending, the reason for this is that the SUMO federate stops before the *onShutdown* function of the vehicle applications is called. Therefore, no data can be requested from the TraCI API when this happens.

The solution to these problems was to create a separate application with the sole purpose of logging. The server application type is perfect for this, as it has no physical presence in the simulated environment and by default does not have any effect on the behavior of entities in the simulation. This data acquisition server is used to log three different values. The first two, which are logged periodically, are the number of vehicles currently in the simulation and the number of vehicles currently charging. The final value is tracked throughout the entire simulation and logged at the end. This is a list of all vehicles that have reached their destination that can be used to filter the results.

Figure 4.5: **SUMO road network used during simulation. Created from OSM data of area around the Porsche Engineering office.**

4.2.5 Simulated Environment

The simulated environment is defined by all the data and configurations set in the MOSAIC scenario. The first step in setting up a scenario is creating a SUMO network. Since the simulation would eventually be linked to the test vehicles it makes sense to base the network on the area in which the test vehicle would be used. This is because the test vehicle would use its GPS location to compare the different charging stations to each as it has no knowledge of the SUMO network structure. Fortunately, the developers of SUMO created tools that can convert map data from OSM to a SUMO network. The size of the network was chosen be a 10x10km area. A large network is more interesting as it would make it possible to have longer trips in which en-route charging would be more relevant. However, the network has a big impact on simulation speed, so a smaller area was chosen. The network that was created based on the OSM data, and is used for evaluation of the system, is shown in Figure 4.5.

Vehicles in SUMO need to have a predefined route that they will follow, which is provided when they are inserted to the simulation. Therefore, routes would have to be generated before the simulation. This also means that across multiple runs of the simulation the same routes would be used, which helps with creating an equal comparison between different runs. The optimal solution would be to have routes based on real-life traffic data in order for the traffic in the simulation to be as realistic as possible. However, this data is not publicly available for the area that was selected. The next best option is to generate random routes in the network using the script that comes with SUMO. This would also make it easy to rebuild the routes if a different area was chosen to create the SUMO network from. Using the script 100 unique routes were generated that favored using points on the edge of the network as the start and end points to emulate through traffic and create longer routes on average.

Charging stations are implemented as parking areas in the SUMO network as discussed in Section 4.2.2. These are not imported from the OSM data, which would also not be desirable, and therefore have to be inserted manually into the network using the netedit tool. A total of 10 charging stations were inserted and spread approximately uniformly through the network. Each of these charging stations has 5 charging spots, each with a charging power of 48kW. These values were chosen such that the total amount of charging power available would be enough to service the charging demand of all the vehicles in the simulation, and thus theoretically allow a scenario with no waiting times, but would not have that much excess that waiting times would always be non-existent.

The next part of the scenario definition is the mapping configuration. This defines which vehicles are spawned, where they are spawned, and how many are spawned in the scenario. In the documentation on the electric vehicle model implemented in SUMO it is stated that the accuracy of the vehicle consumption is only accurate under realistic traffic conditions [60]. Because of this the goal was to get as close as possible to a realistic level traffic. In order to determine the maximum number of vehicles the simulation could handle, tests were performed with increasing number of vehicles spawned on each of the 100 routes. The results of these tests can be seen in Figure 4.6 and it shows a rapid increase in the average journey time of a vehicle when going past 30 vehicles per hour for a single route. It is expected for the journey time to increase when there are more vehicles on the roads as vehicles will encounter more traffic thus slowing down the average travel speed. However, this rapid and big increase was not expected. After examining the simulation it was found that this was caused by inaccuracies in the routes and the network. The conversion from OSM data to a SUMO network makes use of some heuristics to perform actions such as joining junctions. This is done for example when a road with a tram line splitting lanes crosses another road. From the OSM data directly this would create two separate intersections, while this is not desirable for simulation purposes. The developers of the conversion tools recommend the use of some of these settings when converting from OSM to SUMO, however there are still situations in which the network is not imported correctly. These faulty junctions can lead to deadlocks that drastically increase journey time as vehicles will only teleport to the next edge after being blocked for a longer period of time. Problems can also arise with the use of random routes as they could sometimes start or end with a turnaround in the middle of a road because the direction of the route does not match the direction of lane they are spawned on. This can also cause big traffic jams as vehicles wait for oncoming traffic to pass before making the turn. Although the problem with routes can be circumvented with some settings in the generation script these problems could still occur unless the problem points in the network are manually fixed. Therefor, the number of vehicles for each route is limited to 30 per hour. With these traffic conditions the reported consumption levels seem to match what was expected by the developers of the battery model.

After having defined the number of vehicles and their start locations the final

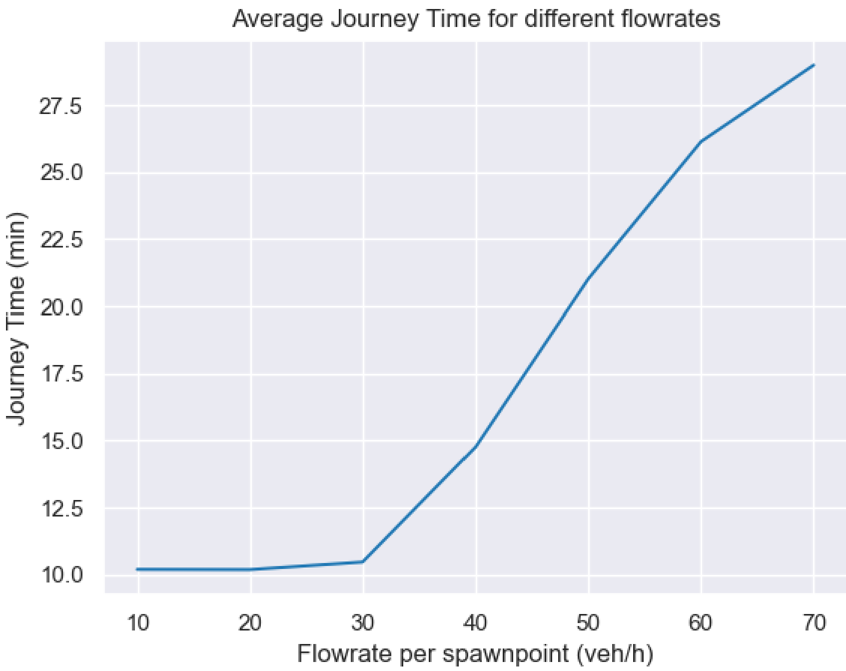

Figure 4.6: **Average journey time based on flowrate at spawning points. Showing significant network congestion at flowrates higher than 30 vehicles per hour.**

step is to define a distribution for vehicle types. The vehicle population consists of both EV and gasoline-powered vehicles. This is done in an attempt to keep the simulated scenario close to a realistic scenario, and comes with the added benefit that the gasoline-powered vehicles do not require any application code and thus improve the performance of the simulation. The share of EVs is set to 10%, which matches current statistics for the global EV share found in [36]. The distribution between the different EV applications discussed in Section 4.2.4 varies depending on the experiment and will be further discussed in Chapter 5.

An important aspect of the simulation is reproducibility. There are multiple instances where random number generation is used to determine what happens in the simulation. It is used for the time between spawning of vehicles on each of the routes, which type of vehicle is spawned, as well as the SoC parameters in the EV application. Especially during development it is important to have consistency between simulation sessions, which is why MOSAIC provides the option to give a seed for its random number generator. The MOSAIC random number generator is then also used inside the EV application to set the SoC parameters, giving the option to perform multiple simulations under the exact same conditions.

4.3 Test Vehicle Integration

To integrate the simulation environment with the test vehicle three systems need to be modified. The simulation environment needs to communicate with the V2X hardware through UDP sockets. The V2X hardware should be able to transmit and receive EVCSN and EV-RSR messages, which are not included in the default software. Finally, an application has to be created on the test vehicle to process the V2X data, visualize it, and handle user input to select a charging station. All algorithms used on the test vehicle should match the ones used in the simulation such that the test vehicle acts identical to the simulated vehicles, effectively increasing the number of vehicles in the test by one.

In the simulation, two UDP sockets are created. The first is shared across all charging station applications, and is used to transmit EVCSN data. This happens every time the charging station broadcasts a message in the simulation, and includes both the static and dynamic information of the EVCSN. To accept data coming from the V2X hardware a new application is introduced. This application is of the server type and has a UDP socket taking data from EV-RSR messages received by the V2X hardware. Once an EV-RSR message is received an interaction is sent to all of the charging station applications. Using an identifier in the interaction, the charging station that the EV-RSR is targeting handles the reservation and a confirmation message is sent back to be transmitted from the V2X hardware to the test vehicle.

On the V2X hardware the basic application is extended to support EVCSN and EV-RSR messages. The architecture of the hardware uses different threads to handle scheduling of messages, transmitting, and receiving. The new messages are implemented to only be sent when new data is received from the UDP connections between hardware and either the simulation or the JUPITER vehicle. This implementation is independent of whether the hardware is connected to the laptop running the simulation or to the vehicle PC, as each of these send different types of messages. The simulation side sends EVCSN, EV-RSR reservation response, and EV-RSR cancellation response messages. Whereas the JUPITER side sends EV-RSR reservation request and EV-RSR cancellation request messages. A periodic event is set in the scheduling thread that is handled by the transmitter thread to check for new data. If new data is available the message is constructed, encoded, and consequently sent. The receiving thread periodically checks if any new messages have been received and decodes them. The relevant contents of the message are then transmitted through UDP to the connected device. As mentioned in Section 3.3 the message definitions are provided to the V2X hardware in the ASN.1 language. The definition used for the EV-RSR is found in Appendix B, and the EVCSN definition in Appendix A with the modifications shown in Listing 4.1.

The JUPITER development platform is created for the validation of ADAS projects, and has a Linux PC onboard that is connected to the internal vehicle data bus. Most of the data that is available on this bus, coming from the different sensors on the car, is published on ROS topics by a ROS bridge node. The data that is relevant for the selection of a charging station are the GPS location and the SoC of the battery. The JUPITER vehicle that was available for this project is a Cayenne, shown in Figure 4.7, which is not a fully electric vehicle. However, since it is a hybrid model there is still SoC data available. The only difference between this vehicle and a full EV, for example the Porsche

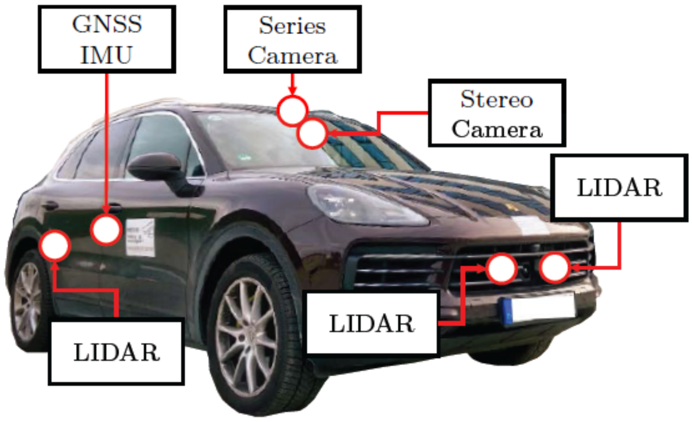

Figure 4.7: **The JUPITER development platform, based on the Porsche Cayenne [32].**

Taycan, is that the battery capacity is significantly smaller and that it would require different models for energy consumption. However, for the purpose of the demonstration the vehicle would be stationary, so the energy estimation done in the simulation to determine if the vehicle is able to reach each of the charging stations is omitted from the implementation. In addition to the car data, taken from the ROS topics, the implemented ROS node also opens a UDP connection to the V2X hardware to receive the EVCSN data, and send data for the V2X hardware to construct EV-RSR reservation request and EV-RSR cancellation request messages.

In order to show the user all the available information and allow them to select a charging station a GUI is created. The GUI that is used is shown in Figure 4.8. Here the static and dynamic data of all 10 charging stations in the simulation is shown. For each of the charging stations it shows the GPS location, the driving time and driving distance from the current location to the charging station, and finally the current waiting time and expected waiting time on arrival. The expected waiting time is computed in the same way as done by the vehicles in the simulation and can be seen in Listing 4.2. Besides data related to the charging station the current SoC of the vehicles is shown at the top. This will be used when a charging intention is shared to determine the required charging time. Finally, on the right hand side the current GPS location of the vehicle is given as well as the option to select a destination. The application is not directly connected to the infotainment system of the vehicle and therefore does not have access to the navigation module. The GUI allows for the selection of four preset destination in the area of the Porsche Engineering office to show how the system adjusts its recommendation to the current route. Using the current location, expected waiting times at the charging stations, and the set destination a recommendation is made by the system based on which charging station would add the least total time the journey and highlights that charging station in blue. By clicking on any of the charging station buttons a selection can be made for that station, which will trigger an EV-RSR reservation request transmitted from the JUPITER V2X hardware to the V2X hardware

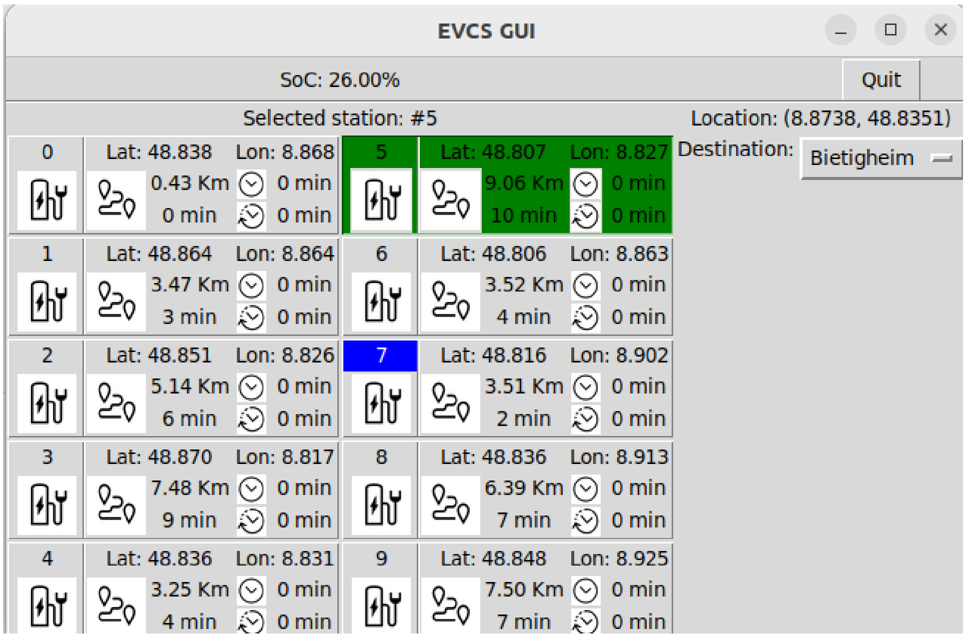

Figure 4.8: **GUI for JUPITER test vehicle. Showing data on charging stations, the currently chosen charging station (green), and the systems recommendation (blue).**

connected to the simulation laptop. When the EV-RSR request is processed a reservation response message is sent back and the selected charging station will be highlighted in green showing that the charging intention has been received. If at any time a different charging station is selected a cancellation request is first sent to the charging in the simulation. Only after a cancellation response has been received will the application attempt to send a new reservation request.

Chapter 5

Evaluation

In this chapter the simulation results are shown, and the different strategies introduced in Section 3.2 are compared against each other. The chapter is structured as follows. First the workflow of the simulation process is explained. After that a baseline is set using the optimal scenario of all charging stations having infinite capacity. Then, values for the frequency of EVCSN messages, and the number of future vehicles included in an EVCSN message are determined. Once all parameters are set the different strategies are compared against each other to show which performs the best at minimizing the total journey time. Finally, experiments are discussed where the fraction of EVs equipped with V2X technology is gradually increased.

All simulations are performed on a laptop with a Intel i7-11850H CPU with each core running at 2.50GHz and 32GB RAM, however the simulation time is limited by the single core performance as the SUMO simulator cannot be parallelized and takes the most computation time. Each simulated scenario simulates 12 hours where vehicles are spawned on each of the 100 routes at a rate of 30 vehicles per hour within a Poisson distribution, of which 10% are EVs. The initial state of each charging station is that all charging spots are empty. For the experiments to determine EVCSN parameters each scenario is repeated 5 times, for all the following experiments each scenario is repeated 10 times. A total of 251 simulations were performed to gather all the results, taking a total of 130 hours of computation time to simulate 117 days. From all these simulation a total of 1.4 Terabytes of raw data was generated.

5.1 Scenario evaluation

In order to analyze the effect of the different strategies on the total journey time the scenario needs to be set up as such that the choice of charging station to use has a significant effect on the total journey time. In Chapter 3 it was mentioned that the waiting time is the component where the most difference could be made. Therefore, the scenario that is used for evaluation should have enough capacity to service the charging requirements of all vehicles, such that in an ideal scenario there would be no waiting time. In order to assess whether this is the case a series of experiments is performed where each of the charging stations in the simulation has infinite capacity, allowing all vehicles to visit the

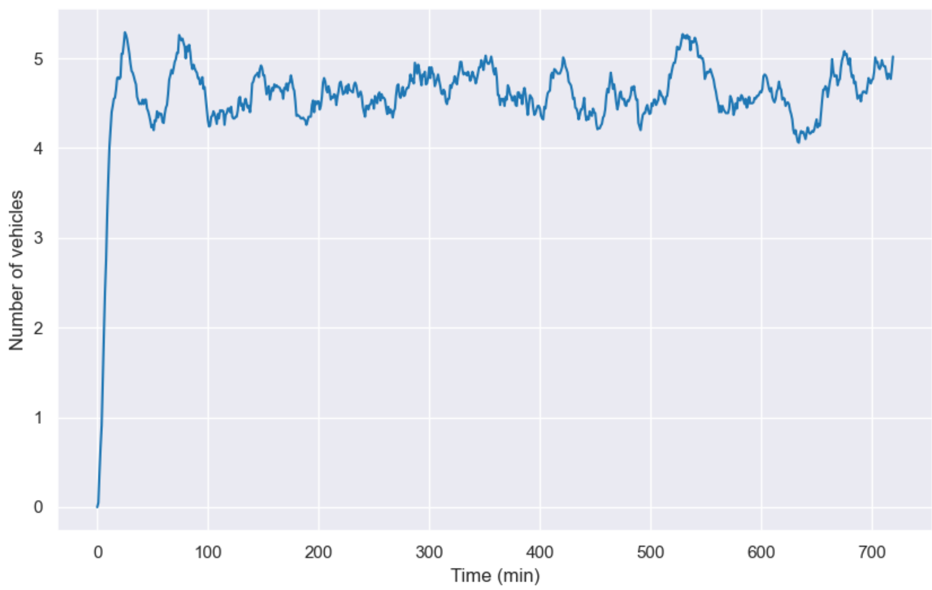

Figure 5.1: **Average number of vehicle charging at a charging station during a simulation where charging stations have infinite capacity.**

charging station that is closest to its route. The expected result would be that the average number of concurrent vehicles charging at each station does not exceed the capacity of a charging station. This would indicate that all charging demands can be met by the charging stations in the simulated area. A rough estimation of the charging demand from the incoming flow of vehicles showed that equipping each charging station with 5 charging spots that each have 48KW of charging power should be sufficient.

The results of this experiment are shown in Figure 5.1. As can be seen in the graph the average number of vehicles charging does not exceed 5 except for some very short moments. This indicates that the charging supply and demand in the simulation is well-balanced, which was the goal. The variation of the number of vehicles charging at a time is quite high, which can be explained by the large effect of randomness involved in the simulation. Random number generation has effect on vehicle spawn intervals, whether the spawned vehicle is an EV, the initial SoC of the vehicle, and the threshold SoC. This can, in the short term, cause a large difference in charging demand. However, over a longer period of time this balances out. If the supply and demand were not balanced, or if scenarios should be evaluated where demand is greater than supply this can be achieved by adjusting one of the following: charging power of charging stations, number of charging spots at a charging station, share of EVs in the simulation, starting SoC, and threshold SoC.

5.2 Establishing Parameters

Besides parameters related to the simulation mentioned in the previous section, there are some parameters pertaining to the EVCSN messages that need to be

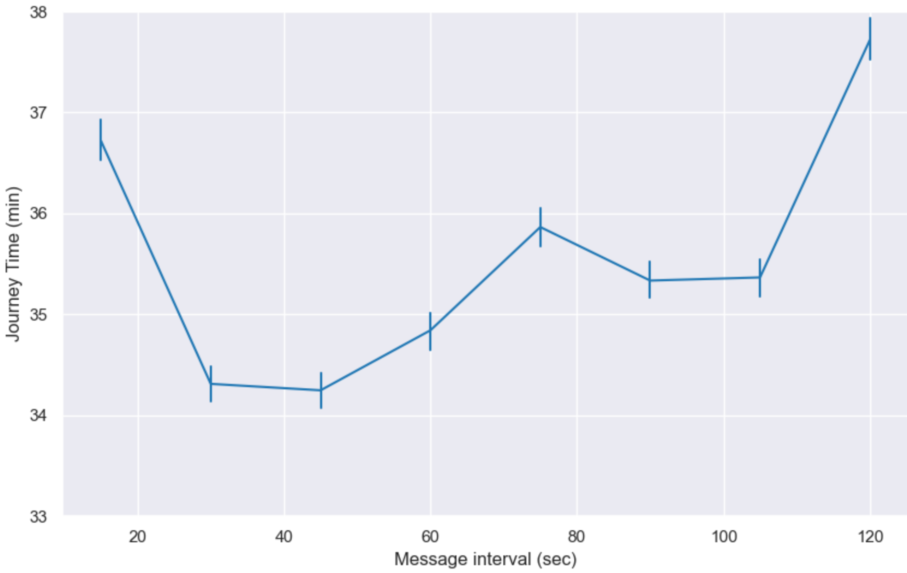

Figure 5.2: **Average journey times for vehicles at different EVCSN message intervals.**

set. These are the frequency at which charging stations send EVCSN messages, and the number of shared charging intentions to include in the message. Both of the parameters can have an impact on the performance of the V2X based strategies. The expectation is that vehicles will be able to achieve lower total journey time with a high frequency of EVCSN messages, and a large number of charging intentions included in the message. In both of these cases the vehicle has access to more data available to it when making its decisions and should therefore allow it to make a better decision. However, increasing both these parameters comes at the cost of increased bandwidth usage. Therefore, experiments are performed to determine appropriate values for these parameters. For the experiments all EVs will be equipped with V2X technology, make use of EV-RSR messages, and continuously re-evaluate its charging station choice while driving.

In Figure 5.2 the effects are shown of increasing the interval at which charging stations send EVCSN messages. In contrast to expectations, the journey does not linearly increase with the message interval. Instead, there seems to be a quadratic relation between the two, with an optimum around the intervals of 30 to 45 seconds between EVCSN messages. A theory for why going from an interval of 15 seconds to 30 seconds between messages has such a significant impact on journey time is that when intervals between messages are too low the system is unstable. Since vehicles react to each other as charging intentions get shared through EVCSN messages and they continuously re-evaluate the best choice it could happen that a pair of vehicles constantly changes their route in reaction to new information, causing an increase in journey time. Whereas, if there is more time between messages changing the route could no longer be the best option since it has already been moving in the direction of the initial choice. Since no significant difference is seen between an interval of 30 and 45 seconds

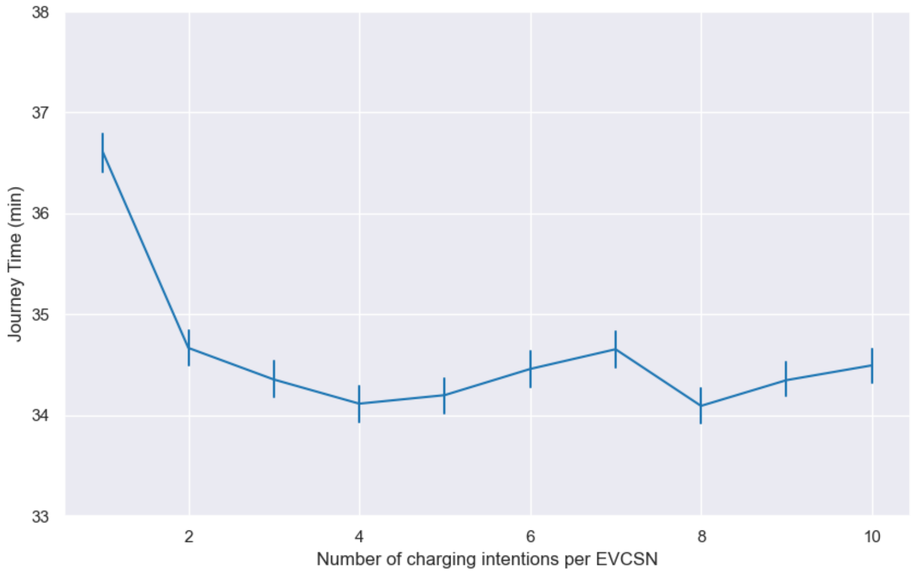

Figure 5.3: **Average journey time for different number of charging intentions shared per EVCSN.**

an interval of 45 seconds was chosen to be used for any future experiments. The higher interval of the two was selected to minimize the number of EVCSN messages that need to be sent, which increases simulation speed and would put less strain on the V2X network in a real-life scenario.

The result for different number of charging intentions included in each EVCSN message shown in Figure 5.3 are in line with expectations. Having more information available about how many vehicles plan to visit a charging station allows vehicles to make a better estimation of waiting time at different charging stations leading to lower overall journey times. However, the effect of having more than two intentions in each message does not seem to be significant. A likely reason for this is the size of the simulated area. The average driving time for the vehicles is only 11 minutes, which means that vehicles cannot share their charging intentions well in advance, and therefor only a small number of intentions for a station are available. When the simulated area is increased the time between sharing the intention and arriving at the charging station should increase and the differences between the number of charging intentions included in a message should increase. For all future experiments a maximum of 10 charging intentions are included in an EVCSN message. The effects on message size from increasing the maximum number charging intention included are minimal, as each charging intention consists only of a pair of timestamps. On top of that, if the list cannot be fully filled it will be shortened and not stuffed with empty entries.

The plots showing the results of the experiments both show the 95% confidence interval for each of the data points. These are consistent for all different values of the parameters and in line with results from the optimal scenario.

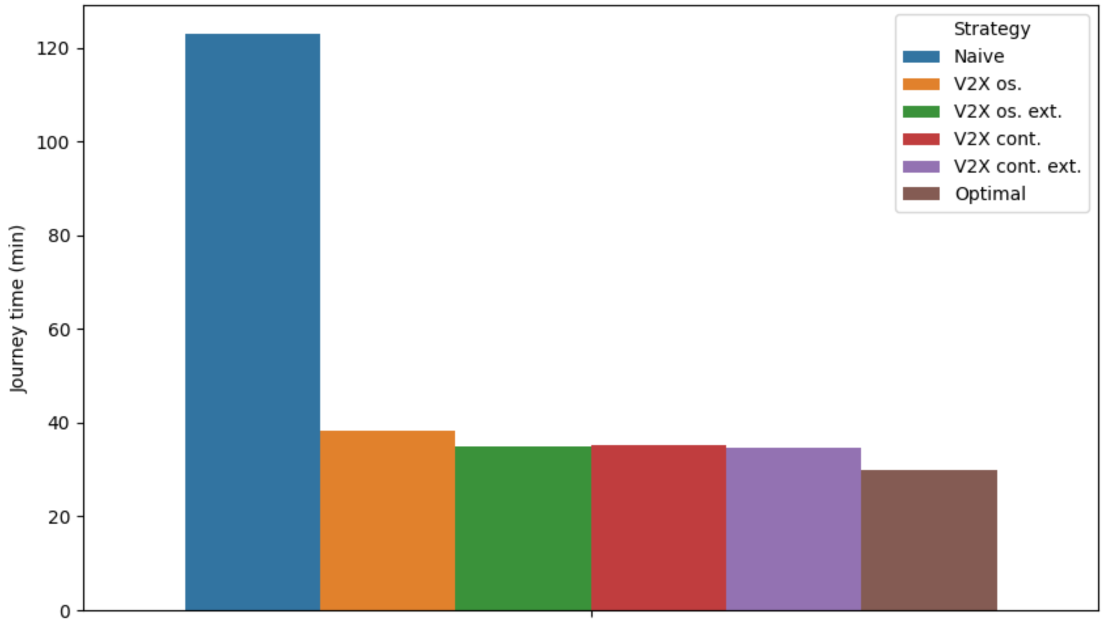

Figure 5.4: **Average journey times for different charging station selection strategies. Showing strategies using V2X communication significantly outperform the naive scenario.**

5.3 Comparing strategies

Now that appropriate values for the interval of EVCSN messages, as well as the number of charging intentions to include in the messages has been established, it is time to compare all the different strategies to each other. The four different selection strategies, based on V2X communication, outlined in Section 3.2 are compared against each other, as well as the optimal scenario where each charging station has infinite capacity, and a naive scenario where vehicles do not use V2X communication. For each of the experiments all of the EVs in the simulation will use the same strategy, the remaining parameters are all kept the same across experiments.

The results of performing 10 instances of a 12 hour simulation run are shown in Figure 5.4. From this it is immediately clear that any strategy that involves V2X communication outperforms the naive scenario significantly. From this it can be observed that some charging stations are positioned in such a way that much traffic passes it naturally. Whereas others are slightly away from commonly traversed roads and would thus not be visited if vehicles only aim to minimize extra driving time. By simply having knowledge of the current waiting times at the charging stations, vehicles can spread themselves much more evenly across charging stations causing a reduction in overall journey times. Additionally, these experiments show that having access to information about the intentions of other vehicles causes a reduction in the overall journey time. Finally, the differences between knowing the intentions of other vehicles and constantly re-evaluating the optimal charging station are very small.

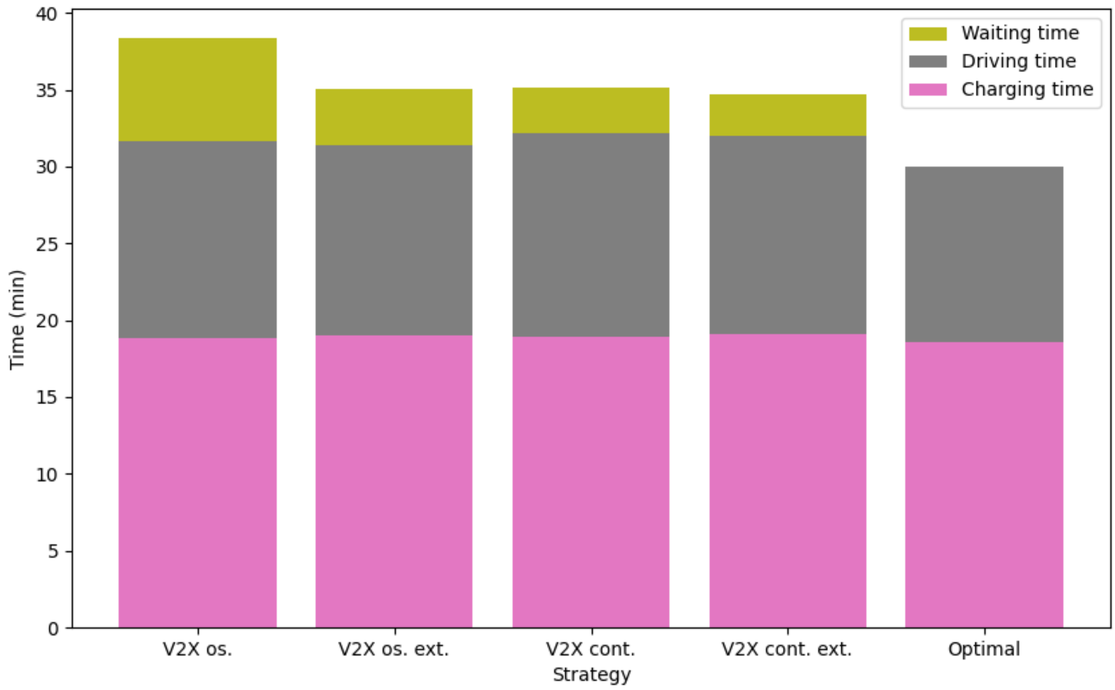

Figure 5.5: **Breakdown of average total journey time of V2X based strategies into separate components. Showing how the V2X based strategies achieve a similar journey time in different ways.**

Figure 5.5 shows the results from the previous experiment in a slightly different format. In this figure a zoomed in view of the four V2X based strategies as well as the optimal scenario is shown. In addition to zooming in on these experiments the total journey time is broken down into its three components: charging time, driving time, and waiting time. From this it shows that while the differences between the strategies V2X OS. EXT., V2X CONT., and V2X CONT. EXT. are very small, they achieve that result in different ways. Across all different strategies the differences in charging times are not significant, as this is mostly determined by the initial SoC of the vehicle as it enter the simulation. The difference is found in the distribution between driving time, and waiting time. When the choice of charging station is periodically re-evaluated it results in longer driving times on average. This is because as soon as a charging station is selected the vehicle will start to drive towards that particular station, and in doing that possibly away from other stations. If, then later, based on new information, another charging station is selected as the better option the route to get there could involve backtracking. However, by doing this, lower average waiting times are achieved. Having access to the charging intentions of other vehicles in addition to re-evaluating charging options periodically reduces the total journey time slightly, however the difference between this strategy and V2X OS. EXT. is less than 30 seconds on the total journey of around 30 min ($< 1\%$).

In [55] the authors reported that the waiting times are minimal if vehicles are evenly spread across the available charging stations. This not only serves the benefit of the vehicles as they wish to minimize waiting and total journey time, but is also beneficial for charging station providers, and grid operators. Figure 5.6 shows how well the different charging station selection strategies achieve this. It shows the average number of vehicles at the charging stations, along with a 95% confidence interval. The data for the naive and optimal scenario show what was already previously observed. Selecting the charging station that is closest to the route of the vehicle results in a very skewed distribution of vehicles across charging station, with a small number of charging station getting the most traffic. The V2X based strategies are shown to be much better at spreading vehicles across the charging stations. This data also nicely shows the difference between the strategies that periodically re-evaluate their charging stations and those that do not. The lower waiting times achieved by strategies periodically re-evaluating their decision achieve a better spread across charging stations, shown by the tighter confidence interval.

5.3.1 V2X Penetration Rate

In order for V2X based strategies for selection of charging stations to be widely adopted there needs to be an incentive for drivers to use it. People are generally selfish, and wish to obtain the best results for themselves, irrespective of how that affects the average result. Therefore, in order for V2X based strategies to be adopted they need to outperform scenarios without V2X independent of how many people are using V2X. This experiment is the only one where the population of EVs is not homogeneous. A portion of the EVs population is equipped with V2X technology and uses the V2X CONT. EXT. strategy. Whereas, the remainder of the vehicles does have V2X technology. The share of the total population using V2X is referred to as penetration rate of V2X technology.

The results of these experiments are shown in Figure 5.7. From this two important observations can be made. First of all, as the V2X penetration rate increases, the average journey time decreases. Secondly, an individual vehicle is always better of, in terms of journey time, by having V2X technology as opposed to not having it. What is interesting to note is that as the penetration rate of V2X technology goes up, the average journey time of vehicles equipped with V2X trends up slightly. This has to do with the fact that in the naive scenario some charging stations go completely unused, and if only a small portion of vehicles knows this through V2X messages they achieve almost zero waiting time at the cost of a little extra driving time. As the V2X penetration rate continues to go up, more and more vehicles will see this discrepancy between charging stations and waiting times will start to occur at less visited charging stations as well.

5.4 Discussion

The results of the experiments covered in the previous sections show promising results for the use of V2X technology to improve en-route charging for EVs. However, there are still some limitations to the simulation setup and the performed experiments that could have an impact on the accuracy of the results.

The first is the degree of realism. This is mostly present in the traffic conditions that are used in the simulated environment. Due to a lack of available data the choice was made to use randomized routes. Also, since no real life data was used the traffic flows are constant. Whereas in reality, these would be highly dependant on the time of day and even the day of the week. With these non-constant flows the most important thing will be how it the incoming flow relates to the capacity of the charging stations. In the examined scenario these were set to be about equal. It is expected that in a case where demand is significantly lower than the charging capacity the differences between using V2X and not will be smaller, and if the demand is significantly higher than the capacity the differences will increase. An effect of using randomized routes is that traffic should be evenly distributed among the road network, since each part of the area has an equal chance of being a start or end point of a route. In reality however, since people do not travel to random locations, there are more significant differences in busyness between different area. This is likely to cause a larger number of vehicles to pass the same charging stations along their optimal routes. Therefore, it is expected that this will increase the differences in performance between the V2X based strategies and the naive scenario.

Secondly, there is the size of the simulated area. Due to technical limitations the simulated area was limited in size to 10x10km. This is quite small for the scenario that was being researched. A typical modern EV has a range of a couple hundred kilometers. So even in a case where the vehicle departs for its journey with a partially charged battery it is unlikely that en-route charging would occur in an area where the start and end of the journey are within a 10x10km area. Increasing the size of the simulation however, comes with some challenges. First and foremost, the simulation would become more computationally intensive. Meaning that collecting results would require either more time, or a more powerful computer. Besides that, the way charging station data is added to an EVCSN would also need to be reconsidered. It is reasonable to assume that a charging station would advertise its state within a 10km range using a EVCSN message. However, if the simulated area would increase to say 100x100km then this same assumption would no longer hold. This would introduce new variables to the simulation related to the transmission range of EVCSN, which would also increase the effect on journey times of placement of individual charging stations in the area. Additionally, as the average length of a journey increases along with the simulated area. This would also give vehicles more opportunity to share their intentions in advance and give them a longer window for finding an optimal charging station before reaching the destination. These changes would hopefully highlight the differences between the V2X OS. EXT., V2X CONT., and V2X CONT. EXT. strategies, which in the current evaluations have very similar performances.

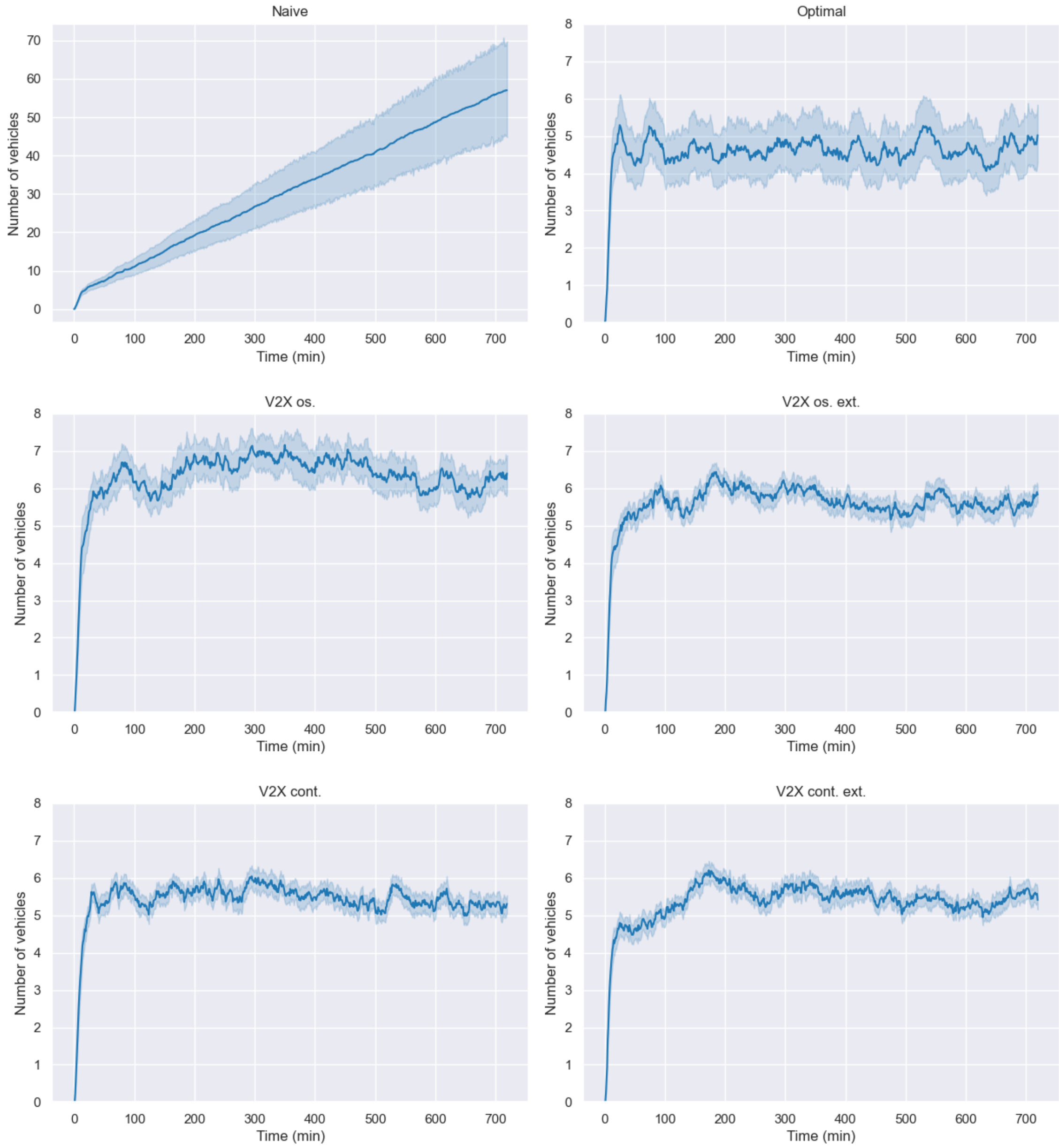

Figure 5.6: **Average number of vehicles charging at a charging station over the duration of the simulation. A constant number of vehicles at a charging station and a tight confidence interval indicate the most equal spread across charging stations.**

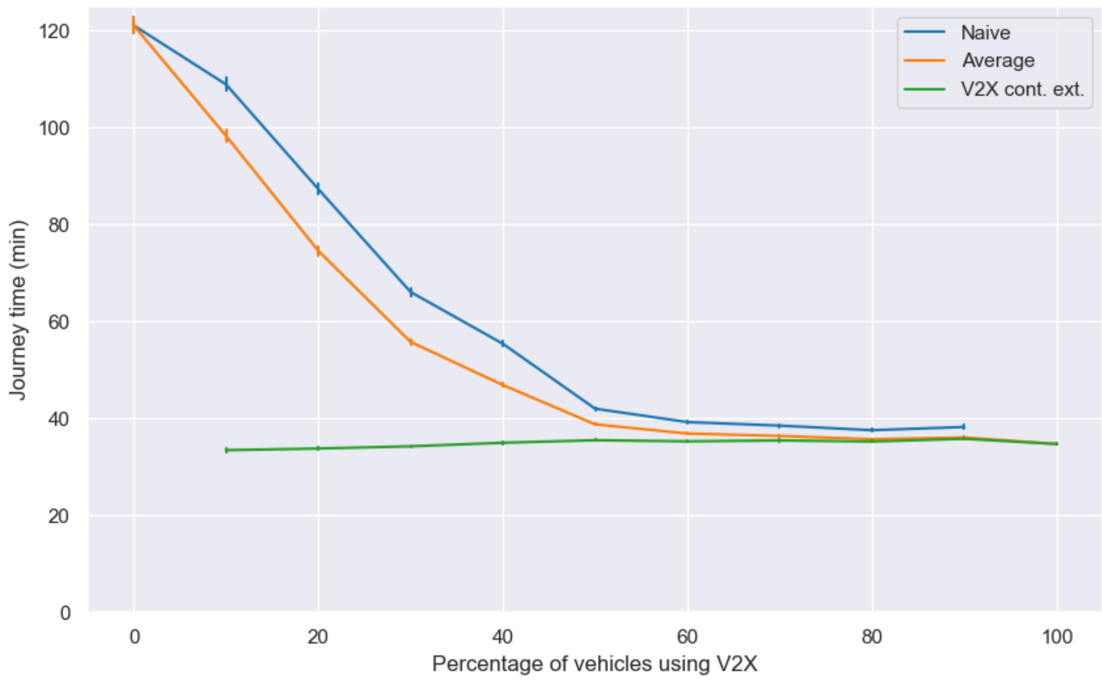

Figure 5.7: **Average journey times for different V2X penetration rates. Showing that an individual vehicles is always better of having V2X than not, and that more vehicles having V2X is better for the total vehicle population.**

Chapter 6

Conclusions

In this thesis project a simulation environment has been created that can be used to evaluate intelligent transport systems using V2X in large-scale traffic scenarios. The simulation setup consists of multiple state-of-the-art simulators, each handling the simulation of a specific sub-problem, coupled together using the MOSAIC co-simulation framework. A realistic road network was created with the use of open source map data from OSM, and a mix of electric vehicles and gasoline-powered vehicles were added that follow randomly generated routes.

Using this simulation environment various strategies to minimize journey time for en-route charging were evaluated. These strategies rely on the use of the V2X messages defined by ETSI for electric vehicle charging, the EVCSN and EV-RSR messages. In addition to using these messages as specified in the ETSI technical specifications some modifications were introduced. By using the EV-RSR messages to share charging intentions of vehicles to the charging station and consequently including these shared charging intentions in the EVCSN message sent by the charging station vehicles can be made aware of charging behavior of other vehicles in advance. Using this information, a vehicle can more accurately estimate waiting times at various charging stations and more effectively minimize its journey time.

In addition to evaluating these strategies in a simulated environment a proof of concept was shown on a test vehicle equipped with V2X communication technology. The test vehicle was able to observe the state of charging stations in the simulation through real EVCSN messages transmitted by C-V2X hardware, in addition to sharing its intention to charge at a specific station specified by the driver through a GUI.

When evaluating the performance of the different strategies, comparisons were made between a naive scenario, where vehicles do not have access to V2X technology to receive messages detailing the dynamic state of the charging stations, and an optimal scenario, where each charging station has infinite capacity. The results showed that strategies using V2X communication significantly outperformed the naive scenario, and approached the optimal scenario. Between the V2X based strategies a difference was noted that knowing how the state of the charging state changes over time, either by knowing about vehicles planning to visit the charging station or periodically re-evaluating the current state of the charging stations, achieves a lower total journey time.

An analysis was also done to establish an appropriate interval for the EVCSN message. In contrast to expectations this showed an optimum at 30-45 seconds between EVCSN messages instead of the expected linear relation, where a smaller interval would result in lower total journey times. This shows that an overload of information can cause instability in the system and result in a deterioration of performance.

Finally, in order for this system to be successfully introduced in production vehicles there needs to be an advantage to its users. This advantage was shown by performing experiments with increasing penetration rates of V2X equipped vehicles in the vehicle population. The results showed that a vehicle using V2X technology will always have a lower average journey time than vehicles that are not using V2X technology. Additionally, it also shows that the largest advantage for V2X equipped vehicles, when compared against those who do not use V2X, is achieved at lower penetration rates. This incentivises users to adopt the technology early to gain the most benefits.

Chapter 7

Future Work

There are two main directions for future work on this topic. The first, and most important, is to perform more experiments to quantify the performance of the V2X based charging station selection strategies. For these additional experiments the realism of the simulated environment should be improved to more closely match the scenario of en-route charging. With the current simulated area of 10x10 Km the average route driven by a vehicle is relatively short. Whereas en-route charging happens mostly in cases where the distance between origin and destination is large. This would give vehicles the opportunity to share their charging intentions further in advance and by doing that allow other vehicles to make more accurate estimations of the dynamic state of the charging stations. Another improvement would be to use realistic traffic data instead of randomly generated routes. This new scenario can then be used to give better insights into the differences between the different V2X based strategies, which currently perform very similar.

The second direction that can be taken is to implement a more realistic model for V2X communication. In the current implementation V2X communication is overly simplified. It is assumed that entities are able to communicate with each other independent of their relative positions. While this is possible with the use of uplink-downlink communication, this type of communication is still in early stages of development and most focus has been placed on PC-5 sidelink communication, where the communication range is much smaller (<1 Km). A setup with a V2X model using only sidelink communication would give insights into the amount of additional infrastructure that is required to make such an intention sharing system successful, or give opportunities to evaluate the performance of VANET based solutions for transmitting messages by hopping from vehicle to vehicle until the destination is reached. In this case it is especially interesting to see the effects of V2X penetration rate and vehicle density required to successfully transmit messages.

In addition to the directions mentioned above, the strategies could also be modified to allow minimization/maximization of different metrics. Users could for instance have a preference for a shorter waiting time at the charging station, even if that results in a longer total journey time. Or, in a case where charging stations have dynamic pricing based on demand, a vehicle can plan its route such that charging costs are minimized. This would give users a larger freedom to personalize their journey to their preference.

Bibliography

[1] 3GPP. 3rd generation partnership project; technical specification group services and system aspects; enhancement of 3GPP support for V2X scenarios; stage 1 (release 17). Technical report, 3GPP, 2022.

[2] 5GAA. V2X functional and perfomance test report; test procedures and results. Technical report, 5GAA, 2018.

[3] 5GAA. A visionary roadmap for advanced driving use cases, connectivity technologies and radio spectrum needs. Technical report, 5GAA, 2020.

[4] 5GAA. Working group standards and spectrum; study of spectrum needs for safety related intelligent transportation systems - day 1 and advanced use cases. Technical report, 5GAA, 2020.

[5] Maheshwari Adaikkappan and Nageswari Sathiyamoorthy. Modelling, state of charge estimation, and charging of lithium-ion battery in electric vehicle: A review. *Int. J. Energy Res.*, 46(3), 2021.

[6] Sid-Ali Amamra and James Marco. Vehicle-to-grid aggregator to support power grid and reduce electric vehicle charging cost. *IEEE Access*, 7, 2019.

[7] Jaume Barceló. *Models, Traffic Models, Simulation, and Traffic Simulation.* Springer, 2010.

[8] R. Bauza, J. Gozalvez, and M. Sepulcre. Operation and performance of vehicular ad-hoc routing protocols in realistic environments. In *2008 IEEE 68th Vehicular Technology Conference*, Calgary, AB, Canada, 2008. IEEE.

[9] T. Boßmann and I. Staffell. The shape of future electricity demand: Exploring load curves in 2050s germany and britain. *Energy*, 90(2), 2015.

[10] Abby Brown, Jeff Cappellucci, Emily White, Alexia Heinrich, and Emma Cost. Electric vehicle charging infrastructure trends from the alternative fueling station locator: First quarter 2023. Technical report, National Renewable Energy Laboratory, 2023.

[11] Christoph Brunner, Gerda Deac, Sebatian Braun, and Christoph Zöphel. The future need for flexibility and the impact of fluctuating renewable power generation. *Renew. Energy*, 149, 2020.

[12] Yue Cao, Tao Jiang, Omprakash Kawwartya, Hongjian Sun, Huan Zhou, and Ran Wang. Toward pre-empted ev charging recommendation through v2v-based reservation system. *IEEE Trans. Syst. Man Cybern. Syst.*, 51(5), 2021.

[13] Jean-Michel Clairand, Javier Rodriguez-Garciá, and Carlos Álvarez-Bel. Assessment of technical and economic impacts of EV user behavior on EV aggregator smart charging. *J. Mod. Power Syst. Clean Energy*, 8(2), 2020.

[14] European Comission. CO_2 emission performance standards for cars and vans. `https://climate.ec.europa.eu/eu-action/transport/road-transport-reducing-co2-emissions-vehicles/co2-emission-performance-standards-cars-and-vans_en#:~:text=Passenger%20cars%20and%20vans%20(%27,is%20the%20main%20greenhouse%20gas.`, 2023.

[15] Syed Muhammed Danish, Kaiwen Zhang, and Hans-Arno Jacobsen. A blockchain-based privacy-preserving intelligent charging station selection for electric vehicles. In *2020 IEEE International Conference on Blockchain and Cryptocurrency*, Toronto, ON, Canada, 2020. IEEE.

[16] Mathijs M. de Weerdt, Sebastian Stein, Enrico H. Gerding, Valentin Robu, and Nicholas R. Jennings. Intention-aware routing of electric vehicles. *IEEE trans. Intell. Transp. Syst.*, 17(5), 2016.

[17] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, Mountain View, California, USA, 2017. JMLR.

[18] Philip Karl-Heinz Dost, Phillip Spichartz, and Constantinos Sourkonis. Charging behavior of users utilizing battery electric vehicles and extended range electric vehicles within the scope of a field test. *IEEE Trans. Ind. Appl.*, 54(1), 2018.

[19] Fabian Eckermann, Moritz Kahlert, and Christian Wietfeld. Performance analysis of C-V2X mode 4 communication introducing an open-source C-V2X simulator. In *2019 IEEE 90th Vehicular Technology Conference (VTC-Fall)*, Honolulu, Hawaii, USA, 2019. IEEE.

[20] Eclipse MOSAIC - a multi-domain and multi-scale simulation framework for connected and automated mobility. `https://eclipse.dev/mosaic/`.

[21] Jakob Erdmann. Lane-changing models in SUMO. In *Proceedings of the SUMO2014 Modeling Mobility with Open Data*, Berlin, Germany, 2014. DLR.

[22] ETSI. Intelligent transport systems (ITS); infrastructure to vehicle communication; electric vehicle charging spot notification specification. Technical report, ETSI, 2012.

[23] ETSI. Intelligent transport systems (ITS); infrastructure to vehicle communications; part 3: Communications system for the planning and reservation of ev energy supply using wireless networks. Technical report, ETSI, 2014.

[24] Etsi v2x message repository. `https://forge.etsi.org/rep/ITS/asn1?page=1`.

[25] Luis Pieltain Fernandez, Tomás Gomez San Roman, Rafael Cossent, Carlos Mateo Domingo, and Pablo Frías. Assesment of the impact of plug-in electric vehicles on distribution networks. *IEEE Trans. Power Syst.*, 26(1), 2010.

[26] Dino Flore. Initial cellular V2X standard completed. https://www.3gpp.org/news-events/3gpp-news/v2x-r14, 2014.

[27] Directorate-General for Climate Action. Fit for 55: EU reaches new milestone to make all new cars and vans zero-emissions from 2035. https://climate.ec.europa.eu/news-your-voice/news/fit-55-eu-reaches-new-milestone-make-all-new-cars-and-vans-zero-emission-2035-2023-03-en, 2023.

[28] Christoph Goebel and Hans-Arno Jacobsen. Aggregator-controlled EV charging in pay-as-bid reserve markets with strict delivery constraints. *IEEE Trans. Power Syst.*, 31(6), 2016.

[29] 802.11 WG Wireless LAN Working Group. Ieee standard for information technology– local and metropolitan area networks– specific requirements– part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 6: Wireless access in vehicular environments. Technical report, IEEE, 2010.

[30] Cindy P. Guzmán, Nataly Bañol Arias, John fredy Franco, João Soares, Zita Vale, and Rubén Romero. Boosting the usage of green energy for EV charging in smart buildings managed by an aggregator through a novel renewable usage index. *IEEE Access*, 9, 2021.

[31] M.A. Hannan, M.S. Mollik, Ali Q. Al-Shetwi, S.A. Rahman, M. Mansor, R.A. Begum, K.M. Muttagi, and Z.Y. Dong. Vehicle to grid connected technologies and charging strategies: Operation, control, issues and recommendations. *J. Clean. Prod.*, 339, 2022.

[32] Johann Haselberger, Marcel Pelzer, Bernhard Schick, and Steffen Müller. Jupiter – ros based vehicle platform for autonomous driving research. In *2022 IEEE International Symposium on Robotic and Sensors Environments (ROSE)*, Abu Dhabi, United Arab Emirates, 2022. IEEE.

[33] F. Hipolito, C.A. Vanded, and J. Rich. Charging, steady-state soc and energy storage distributions for ev fleets. *Appl. Energy*, 317, 2022.

[34] Tyler Hodge. Hourly electricity consumption varies throughout the day and across seasons. https://www.eia.gov/todayinenergy/detail.php?id=42915, 2020.

[35] IEEE. 1516-2010 - ieee standard for modeling and simulation (m&s) high level architecture (hla)– framework and rules. Technical report, IEEE, 2010.

[36] International Energy Agency. Global EV outlook 2023. Technical report, International Energy Agency, 2023.

[37] Daniel Krajzewicz. Kombination von taktischen und strategischen einflüssen in einer mikroskopischen verkehrsflusssimulation. In *Fahrermodellierung in Wissenschaft und Wirtschaft, 2. Berliner Fachtagung für Fahrermodellierung*. VDI-Verlag, 2009.

[38] Stefan Krauß. Microscopic modelling of traffic flow: Investigation of collision free vehicle dynamics. Technical report, DLR, 1998.

[39] Stefan Krauß, Peter Wagner, and Christian Gawron. Metastable states in a microscopic model of traffic flow. *Phys. Rev.*, 55:5597–5602, 1997.

[40] G. Krishna. Understanding and identifying barriers to electric vehicle adoption through thematic analysis. *Transp. Res. Interdiscip. Perspect.*, 10, 2021.

[41] Tamás Kurczveil and Eckehard Schnieder. Implementation of an energy model and a charging infrastructure in SUMO. In *Proceedings of the 1st SUMO User Conference SUMO2013*, Berlin, Germany, 2013. DLR.

[42] Sasha Lekach and Alex Kwanten. Ev survey shows range anxiety igh while charging costs remain low. `https://www.forbes.com/wheels/features/ev-range-cost-confidence-survey/`, 2022.

[43] Nico Lerch. How V2G and V2H can reach scale. `https://www.gridx.ai/blog/how-v2g-v2h-reach-scale`, 2021.

[44] Chu-Hsing Lin, Jui-Ling Yu, Jung-Chun Liu, and Chia-Jen Lee. Genetic algorithm for shortest driving time in intelligen transportation systems. In *2008 International Conference on Multimedia and Ubiquitous Engineering*, Busan, South Korea, 2008. IEEE.

[45] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using sumo. In *The 21st IEEE International Conference on Intelligent Transportation Systems*, Maui, HI, USA, 2018. IEEE.

[46] Pukar Mahat, Kenneth Rønsig Kanstrup, Alberto Palomar Lozano, and Aleksandr Sleimovits. Price based electric vehicle charging. In *2012 IEEE Power and Energy Society General Meeting*, San Diego, CA, USA, 2012. IEEE.

[47] Marc Melliger, Oscar van Vliet, and Heikki Liimatainen. Anxiety vs reality - sufficiency of battery electric vehicle range in switzerland and finland. *Transp. Res. Part D: Transport and Environment*, 65, 2018.

[48] Rafael Molina-Masegosa and Javier Gozalvez. LTE-V for sidelink 5G V2x vehicular communications: A new 5G technology for short-range vehicle-to-everything communications. *IEEE Veh. Technol. Mag.*, 12(4), 2017.

[49] Mosaic sns documentation. `https://eclipse.dev/mosaic/docs/simulators/network_simulator_sns/`.

[50] Ns-2 project documentation. `https://www.isi.edu/nsnam/ns/`.

[51] Openstreetmap. https://www.openstreetmap.org/.

[52] Pc4 travel time matrix for nederland van boven. https://www.geodms.nl/PC4_travel_time_matrix_for_Nederland_van_Boven.

[53] Vehicle-to-x communication: Connected. efficient. safe. https://newsroom.porsche.com/en/2023/innovation/porsche-engineering-vehicle-to-x-communication-connected-efficient-safe-31892.html.

[54] Kejun Qian, Chengke Zhou, Malcolm Allan, and Yue Yuan. Modeling of load demand due to ev battery charging in distribution systems. *IEEE Trans. Power Syst.*, 26(2), 2011.

[55] Hua Qin and Wensheng Zhang. Charging scheduling with minimal waiting in a network of electric vehicles and charging stations. In *Proc. 8th ACM Int. Workshop Veh. Inter-Netw*, Las Vegas, NV, USA, 2011. ACM.

[56] Emmanouil S. Rigas, Sarvapali D. Ramchurn, Nick Bassiliades, and George Koutitas. Congestion management for urban ev charging systems. In *2013 IEEE International Conference on Smart Grid Communications*, Vancouver, BC, Canada, 2013. IEEE.

[57] Julio A. Sanguesa, Vicente Torres-Sanz, Piedad Garrido, Francisco J. Martinez, and Johann M. Marquez-Barja. A review on electric vehicles: Technologies and challenges. *Smart Cities*, 4(1):372–404, 2021.

[58] Karl Schrab, Maximillian Neubauer, Ilja Radusch, Stamatis Manganiaris, Panagiotis Lytrivis, and Angelos J. Amditis. Modeling an its management solution for mixed highway traffic with eclipse mosaic. *IEEE trans. Intell. Transp. Syst.*, 24(6), 2022.

[59] Hanbyul Seo, Ki-Dong Lee, Shinpei Yasukawa, Ying Peng, and Philippe Sartori. LTE evolution for vehicle-to-everything access. *IEEE Commun. Mag.*, 54(6), 2016.

[60] Sumo - electric documentation. https://sumo.dlr.de/docs/Models/Electric.html.

[61] Eve Thomas. Signal: 50 https://www.energymonitor.ai/news/signal-50-of-germanys-power-to-be-generated-by-renewables-in-2023/?cf-view, 2023.

[62] Anna Tomaszewska, Zhengyu Chu, Xuning Feng, Simon O'Kane, Xinhua Liu, Jingyi Chen, Chenzhen Ji, Elizabeth Endler, Ruihe Li, Lishuo Liu, Yalun Li, Sigi Zheng, Sebastian Vetterlein, Ming Gao, Jiuyu Du, Michael Parkes, Minggao Ouyang, Monica Marinescu, Gregory Offer, and Billy Wu. Lithium-ion battery fast charging: A review. *eTransportation*, 1(15), 2019.

[63] Jacopo Torriti. The risk of residential peak electricity demand: A comparison of five european countries. *Energies*, 10(7), 2017.

[64] Stylianos I. Vagropoulos, Dimitros K. Kyriazidis, and Anastasios G. Bakirtzis. Real-time charging management framework for electric vehicle aggregators in a market environment. *IEEE Trans. Smart Grid*, 7(2), 2016.

[65] Axel Wegener, Michal Piorkowski, Maxim Raya, Horst Hellbrück, Stefan Fischer, and Jean-Pierre Hubaux. TraCI: An interface for coupling road traffic and network simulators. In *Proceedings of the 11th Communications and Networking Simulation Symposium*, Ottawa, Canada, 2008. ACM.

[66] Stefan Weitemeyer, David Kleinhans, Thomas Vogt, and Carsten Agert. Integration of renewable energy sources in future power systems: The role of storage. *Renew. Energy*, 75, 2015.

[67] Xiao-Feng Xie, Stephen F. Smith, and Gregory J. Barlow. Schedule-driven coordination for real-time traffic network control. In *Proceedings of the International Conference on Automated Planning and Scheduling*, Saõ Paulo, Brazil, 2012. AAAI.

[68] Jinbiao Xu and Vincent W.S. Wong. An approximate dynamic programming approach for coordinated charging control at vehicle-to-grid aggregator. In *2011 IEEE International Conference on Smart Grid Communications*, Brussels, Belgium, 2011. IEEE.

[69] Shaolun Xu, Zheng Yan, and Donghan Feng. Decentralized charging control strategy of the electric vehicle aggregator based on augmented lagrangian method. *Int. J. Electr. Power Energy Syst.*, 104, 2019.

[70] Shun-Neng Yang, Wei-Sheng Cheng, Yu-Ching Hsu, Chai-Hien Gan, and Yi-Bing Lin. Charge scheduling of electric vehicles in highways through mobile computing. In *2011 IEEE 17th International Conference on Parallel and Distributed Systems*, Tainan, Taiwan, 2011. IEEE.

[71] Ashratuz Zavin, Adnan Sharif, Anika Ibnat, Wali Mohammed Abdullah, and Muhammed Nazrul Islam. Towards developing an intelligent system to suggest optimal path based on historic and real-time traffic data. In *2017 20th International Conference of Computer and Information Technology*, Dhaka, Bangladesh, 2017. IEEE.

Appendix A

ASN1 description of EVCSN message

```
-- Point of Interest (POI) notification for charging spot
-- for electric vehicle
-- EVCSN Message definition
-- ASN.1 Start Definition

EVCSN-PDU-Descriptions {
itu-t (0) identified-organization (4) etsi (0) itsDomain (5)
wg1 (1) ts (101556) evcsn (1) version (1)
}

DEFINITIONS AUTOMATIC TAGS ::=

BEGIN

IMPORTS
    ItsPduHeader,
    StationID,
    TimestampIts,
    ReferencePosition
FROM ITS-Container {
 itu-t (0) identified-organization (4) etsi (0)
 itsDomain (5) wg1 (1) ts (102894) cdd (2) version (1)
};

-- Root Message PDU: EvcsnPdu

EvcsnPdu ::= SEQUENCE {
    header ItsPduHeader,
    evcsn EVChargingSpotNotificationPOIMessage
}

EVChargingSpotNotificationPOIMessage ::= SEQUENCE {
    poiHeader ItsPOIHeader, -- Specific POI Message Header
    evcsnData ItsEVCSNData  -- Electric Vehicle Charging Spot Data Elements
}
```

```
ItsPOIHeader ::= SEQUENCE {
    poiType                 POIType, -- set to "EV charging station POI ID = 1"
    timeStamp               TimestampIts,
    relayCapable            BOOLEAN
}

ItsEVCSNData ::= SEQUENCE {
    totalNumberOfStations   NumberStations,
    chargingStationsData    SEQUENCE (SIZE(1..256)) OF ItsChargingStationData
}

ItsChargingStationData ::= SEQUENCE {
    chargingStationID       StationID,
    utilityDistributorId    UTF8String (SIZE(1..32)) OPTIONAL,
    providerID              UTF8String (SIZE(1..32)) OPTIONAL,
    chargingStationLocation ReferencePosition,
    address                 UTF8String           OPTIONAL,
    phoneNumber             NumericString (SIZE(1..16))     OPTIONAL,
    accessibility           UTF8String (SIZE(1..32)),
    digitalMap              DigitalMap            OPTIONAL,
    openingDaysHours        UTF8String,
    pricing                 UTF8String,
    bookingContactInfo      UTF8String           OPTIONAL,
    payment                 UTF8String           OPTIONAL,
    chargingSpotsAvailable  ItsChargingSpots,
    ...
}

ItsChargingSpots ::= SEQUENCE (SIZE(1..16)) OF ItsChargingSpotDataElements

ItsChargingSpotDataElements ::= SEQUENCE {
    type                ChargingSpotType,
    evEquipmentID       UTF8String OPTIONAL,
    typeOfReceptacle    TypeOfReceptacle,
    energyAvailability  UTF8String,
    parkingPlacesData   ParkingPlacesData OPTIONAL
}

DigitalMap ::= SEQUENCE (SIZE(1..256)) OF ReferencePosition

ChargingSpotType ::= BIT STRING {
    standardChargeMode1(0),
    standardChargeMode2(1),
    standardOrFastChargeMode3(2),
    fastChargeWithExternalCharger(3),
    quickDrop(8),
    inductiveChargeWhileStationary(12),
    inductiveChargeWhileDriving(14)
}

TypeOfReceptacle ::= BIT STRING

ParkingPlacesData ::= SEQUENCE (SIZE(1..4)) OF SpotAvailability
```

```
SpotAvailability  ::= SEQUENCE {
    maxWaitingTimeMinutes INTEGER (0..1400), -- 0 if available or max waiting
                                             -- time (minutes)
    blocking              BOOLEAN            -- true if the spot can be blocked
}

POIType               ::= INTEGER(0..65535)
NumberStations        ::= INTEGER(1..256)

END
```

Appendix B

ASN1 description of EV-RSR messages

```
EV-RechargingSpotReservation-PDU-Descriptions {
 itu-t (0) identified-organization (4) etsi (0) itsDomain (5)
 wg1 (1) ts (101556) ev-rsr (4) version (1)
}

DEFINITIONS AUTOMATIC TAGS ::=

BEGIN

IMPORTS
   ItsPduHeader FROM ITS-Container {
 itu-t (0) identified-organization (4) etsi (0)
 itsDomain (5) wg1 (1) ts (102894) cdd (2) version (1)
};

EV-RSR ::= SEQUENCE {
    header ItsPduHeader,
    messageBody EV-RSR-MessageBody
}

EV-RSR-MessageBody ::= CHOICE {
    preReservationRequestMessage        PreReservationRequestMessage,
    preReservationResponseMessage       PreReservationResponseMessage,
    reservationRequestMessage           ReservationRequestMessage,
    reservationResponseMessage          ReservationResponseMessage,
    cancellationRequestMessage          CancellationRequestMessage,
    cancellationResponseMessage         CancellationResponseMessage,
    updateRequestMessage                UpdateRequestMessage,
    updateResponseMessage               UpdateResponseMessage,
    ...
}

PreReservationRequestMessage ::= SEQUENCE {
    evse-ID         EVSE-ID,
    arrivalTime     TimestampUTC,
```

```
    departureTime   TimestampUTC OPTIONAL,
    rechargingType  RechargingType,
    batteryType     BatteryType OPTIONAL,
    ...
}

PreReservationResponseMessage ::= SEQUENCE {
    preReservation-ID              PreReservation-ID,
    availabilityStatus             AvailabilityStatus,
    preReservationExpirationTime   TimestampUTC,
    supportedPaymentTypes          SupportedPaymentTypes,
    ...
}

ReservationRequestMessage ::= SEQUENCE {
    currentTime         TimestampUTC,
    preReservation-ID   PreReservation-ID,
    arrivalTime         TimestampUTC,
    departureTime       TimestampUTC OPTIONAL,
    eAmount             EAmount,
    eAmountMin          EAmount,
    paymentType         PaymentType,
    payment-ID          Payment-ID,
    secondPayment-ID    Payment-ID OPTIONAL,
    pairing-ID          Pairing-ID OPTIONAL,
    ...
}

ReservationResponseMessage ::= SEQUENCE {
    reservationResponseCode    ReservationResponseCode,
    reservation-ID             Reservation-ID OPTIONAL,
    reservation-Password       Reservation-Password OPTIONAL,
    stationDetails             StationDetails OPTIONAL,
    chargingSpotLabel          ChargingSpotLabel OPTIONAL,
    expirationTime             TimestampUTC,
    freeCancelTimeLimit        TimestampUTC OPTIONAL,
    ...
}

CancellationRequestMessage ::= SEQUENCE {
    reservation-ID          Reservation-ID,
    reservation-Password    Reservation-Password,
    currentTime             TimestampUTC,
    ...
}

CancellationResponseMessage ::= SEQUENCE {
    reservation-ID              Reservation-ID,
    cancellationResponseCode    CancellationResponseCode,
    ...
}

UpdateRequestMessage ::= SEQUENCE {
    reservation-ID          Reservation-ID,
```

```
    reservation-Password    Reservation-Password,
    updatedArrivalTime      TimestampUTC,
    updatedDepartureTime    TimestampUTC,
    ...
}

UpdateResponseMessage ::= SEQUENCE {
    reservation-ID       Reservation-ID,
    updateResponseCode   UpdateResponseCode,
    chargingSpotLabel    ChargingSpotLabel OPTIONAL,
    ...
}

AvailabilityStatus ::= ENUMERATED { available, no-free-capacity }

BatteryType ::= UTF8String (SIZE(1..16))

CancellationResponseCode ::= ENUMERATED { ok, unknown-Reservation-ID,
    mismatching-Reservation-Password }

ChargingSpotLabel ::= UTF8String (SIZE(1..4))

ContractID ::= UTF8String (SIZE(1..24))

EAmount ::= INTEGER { oneWh(1) } (1..500000)

ChargingPower ::= INTEGER { oneW(1) } (1..200000)

EVSE-ID ::= OCTET STRING (SIZE(1..32))

ExternalIdentificationMeans ::= UTF8String (SIZE(1..24))

Pairing-ID ::= VisibleString (SIZE(1..64))

PaymentType ::= ENUMERATED {contract, externalIdentification}

Payment-ID ::= CHOICE {
    contractID                     ContractID,
    externalIdentificationMeans    ExternalIdentificationMeans
}

RechargingType ::= SEQUENCE {
    rechargingMode    RechargingMode,
    powerSource       PowerSource
}

RechargingMode ::=  INTEGER { mode1(0), mode2(1), mode3(2), mode4(3), quickDrop(8),
    inductiveChargingWhileStationary(12), inductiveChargingWhileDriving(14) } (0..15)

PowerSource::=  INTEGER { notApplicable(0), ac1Phase(1), ac2Phase(2), ac3Phase(3),
    dcc(4), chaDeMo(5) } (0..7)

ReservationResponseCode ::= ENUMERATED {ok, invalid-EVSE-ID, payment-type-not-supported,
    payment-error, authentication-error, insufficient-power-availability }
```

```
Reservation-ID ::= VisibleString (SIZE(8))

PreReservation-ID ::= Reservation-ID

Reservation-Password ::= VisibleString (SIZE(8))

StationDetails ::= UTF8String (SIZE(1..32))

SupportedPaymentTypes ::= BIT STRING { contract(0), externalIdentification (1) } (SIZE(2))

TimestampUTC ::= INTEGER { utcStartOf2013(0), oneSecondAfterUTCStartOf2013(1) }

UpdateResponseCode ::= ENUMERATED { ok, unknown-Reservation-ID, mismatching-Reservation-Password,
    invalid-Arrival-Time, invalid-Departure-Time }

StatusResponseCode ::= ENUMERATED { charging-Spot-Occupied, charging-Spot-Unoccupied,
    charging-Ongoing, reservation-Limit-Exceeded, reservation-Cancelled, unknown-Reservation-ID,
    expired-Reservation-ID }

END
```