



UHASSELT

KNOWLEDGE IN ACTION

School of Transportation Sciences

Master of Transportation Sciences

Master's thesis

Smart Awareness System for Drivers (SASD)

Maryam Al Kaabi

Thesis presented in fulfillment of the requirements for the degree of Master of Transportation Sciences, specialization Traffic Safety

SUPERVISOR :

Prof. dr. ir. Ansar-Ul-Haque YASAR

CO-SUPERVISOR :

Associate Prof. Nafaa JABEUR



UHASSELT

KNOWLEDGE IN ACTION

www.uhasselt.be

Universiteit Hasselt
Campus Hasselt:
Martelarenlaan 42 | 3500 Hasselt
Campus Diepenbeek:
Agoralaan Gebouw D | 3590 Diepenbeek

2020
2021



School of Transportation Sciences

Master of Transportation Sciences

Master's thesis

Smart Awareness System for Drivers (SASD)

Maryam Al Kaabi

Thesis presented in fulfillment of the requirements for the degree of Master of Transportation Sciences, specialization
Traffic Safety

SUPERVISOR :

Prof. dr. ir. Ansar-Ul-Haque YASAR

CO-SUPERVISOR :

Associate Prof. Nafaa JABEUR

Smart Awareness System For Drivers

MARYAM AL KAABI

Supervisors:

Professor Ansar Yasar

Professor Nafaa Jabeur

ACKNOWLEDGMENT

First and foremost, I'd like to express my sincere gratitude to my supervisor, Prof. Nafaa Jabeur, for his unwavering support of my M.Sc. study and research, as well as his patience, encouragement, enthusiasm, and vast knowledge, as well as his forbearance of my numerous errors. Your advice was invaluable to me at all times, not just in research but also in everyday life. You provided me with the opportunity to pursue my studies in my chosen field, as well as the opportunity to tap into my research potential, and you taught me how to conduct research and what is critical in the academic world. For my M.Sc. studies, I could not have asked for a better boss and tutor.

I'm also grateful to Dr. Anasr Yasar, who has always been willing to listen and offer guidance. I owe him a debt of gratitude for the lengthy discussions that aided me in figuring out the technical details of my project. I was extremely lucky to have an advisor who allowed me to explore on my own while also providing me with the encouragement I needed to recover when my steps faltered.

I'd like to express my gratitude to my parents for instilling in me a love of artistic pursuits, science, and language, all of which are represented in this study. Thank you, my parents. Thank you for being there to listen and love me, my dear Husband.

TABLE OF CONTENT

Table of Contents

ACKNOWLEDGMENT	1
TABLE OF CONTENT	2
LIST OF FIGURES	3
LIST OF TABLES	4
LIST OF ACRONYMS	5
1. INTRODUCTION	6
1.1. Motivation.....	7
1.2. Research questions.....	7
1.3. Objectives.....	8
1.4. Thesis outline.....	8
2. LITERATURE REVIEW	9
2.1. Intelligent transportation system.....	9
2.2. Smart awareness system / Assistant driver system.....	10
2.3. Simulation Frameworks.....	11
2.3.1. Mobility simulation.....	11
2.3.2. Network simulation.....	12
2.3.3. VANETs simulation.....	15
2.4. Related studies.....	16
3. METHODOLOGY	21
Fundamentals of the proposed solution.....	21
4. SIMULATION SETUPS & SCENARIOS	24
4.1. Part 1: Software setup.....	24
4.2. Part 2 : Open street map (OSM).....	32
5. Scenario results	34
6.1 CONCLUSION & FUTURE WORK	41
Reference:	43
APPENDIX	47

LIST OF FIGURES

FIGURE 1: INTELLIGENT TRANSPORTATION SYSTEM STRUCTURE (WOODROW, 1998).....	9
FIGURE 2 : ARCHITECTURE OF NS3 (JAWANDHIYA, ET AL., 2013).....	13
FIGURE 3 : :OMNET ++ SIMULATOR	14
FIGURE 4: TYPE OF COMMUNICATION IN VANET(HARTENSTEIN & LABERTEAUX, 2010).	16
FIGURE 5 : FLOW CHART OF THE SYSTEM (AGARWAL ET AL, 2018).....	17
FIGURE 6 : INTERACTION BETWEEN NS3 AND MATLAB(KACHAN, 2010).....	18
FIGURE 7 : : SIMULATION EXPERIMENTAL STRUCTURE(CHENXI ET AL., 2012).	19
FIGURE 8 : NS3 AND SUMO INTERACTION THROUGH MATLAB.....	20
FIGURE 9 : BASIC FRAMEWORK SETUP	21
FIGURE 10 :NODE FILE	25
FIGURE 11 : EDGES FILE.....	26
FIGURE 12:ROUTES FILE	28
FIGURE 13: STREET MAP.....	29
FIGURE 14 : TRAFFIC LIGHTS.....	30
FIGURE 15: BLOCK ROAD	30
FIGURE 16:EMISSION	31
FIGURE 17: HASSELT CITY MAP AND SIMULATION IN SUMO	33
FIGURE 18 : TIME LOSS & WAITING TIME	35
FIGURE 19:SPEED & DEPART DELAY.....	36
FIGURE 20 :TRAVELLING TIME WITH SUMO AND VANETS.....	37
FIGURE 21: COLLISION GRAPH	37
FIGURE 22: CO EMISSION	38
FIGURE 23: HC EMISSION	39
FIGURE 24: CO2 EMISSION	39
FIGURE 25: FUEL CONSUMPTION.....	40
FIGURE 26: COMMUNICATION BETWEEN CARS IN VANET	40

LIST OF TABLES

TABLE 1: SUMO STATICS SIMULATION.....	34
TABLE 2 : VARIOUS PARAMETERS OF SIMULATION	35
TABLE 3: COLLISION DATA	37

LIST OF ACRONYMS

ABM	Agent-Based Modelling
API	Application Programming Interface
ATIS	Advanced Traveller Information Systems
ATMS	Advanced Traffic Management Systems
AVCS	Advanced Vehicle Control Systems
ATS	Artificial Transportation Systems
MAS	Multi-Agent System
OSM	Open street Map
SUMO	Simulation of Urban MObility
TraCI	Traffic Control Interface
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
VANETs	Vehicular Ad-hoc NETworks
MAC	Media Access Control
OMNeT++	Objective Modular Network Tested in C++
SASD	Smart Awareness system for drives
ITS	Intelligent transport system
CVO	Commercial Vehicle Operations

Chapter 1

1. INTRODUCTION

Almost half of the world's population now lives in cities, with that number projected to increase to 5 billion by 2030. The recent increase in the number of cars on city roads has resulted in a slew of issues, including traffic congestion, a large number of people killed in automobile accidents, increased fuel usage, and pollution, among others. In today's world, traffic congestion is a major issue in major cities. Congestion and the resulting vehicle accommodation issue are followed by a persistent risk of accidents (Road traffic injuries, 2020).

In 2014, there were more than 2.4 million traffic accidents on German highways, resulting in 392 000 injuries and 3377 fatalities. About 25 700 people were killed in traffic-related collisions in Europe, while more than 200 000 “came home with life-changing, severe injuries”. Many (national and pan-European) programs (for example, Vision Zero) have long aimed to minimize this number and ultimately achieve a traffic system with no serious injuries or fatalities. For example, the National Highway Traffic Safety Administration (NHTSA) estimates that approximately 43,000 people are killed in fatal car accidents in the United States each year (Eurostat, 2020) .

According to annual global road accident figures, road accidents cost the global economy USD 518 billion, with individual countries losing 1-2 percent of their annual GDP. Congestion costs the global economy a lot of money in terms of lost time and electricity. According to a Tom-Tom traffic index that covers 416 cities in 57 countries spanning six continents, 239 cities have seen a significant rise in traffic in 2019 compared to 2018, while only 63 countries have seen a decrease in road traffic. UK commuters, for example, lost an average of 178 hours a year due to traffic congestion in 2018, costing them £7.9 billion. Furthermore, as the number of cars on city roads in most countries grows, cities are at risk of experiencing pollution, collisions, and traffic congestion, both of which contribute to a poor quality of life. Every day, a tremendous amount of fuel, energy, time, and money is wasted due to traffic congestion. For example, in 2011, each driver in the United States spent an additional 38 hours in their car and wasted 19 gallons of fuel due to traffic congestion, resulting in a \$818 congestion cost per auto commuter (Peden, 2016).

To address these issues, countries have begun research projects to find a way to reduce the number of people killed in traffic accidents (MearaJul, 2020). Vision Zero, a Swedish road safety initiative, seeks to reduce road fatalities to zero by 2050. The European Union has set a 50 percent reduction in road deaths by 2020 as an intermediate goal (KHAN, 2020). However, according to data released by the European Transport Safety Council in 2018, there is a significant gap between the desired and real reduction in traffic-related deaths. Since efficient transportation networks are critical to an urban community and serve as its primary communication infrastructure, fast and effective traffic congestion solutions are required to avoid negative effects on the city's social and

economic well being (Hololei, 2021). Using modeling and simulation as a solution to this problem is one choice. A new generation of mobility systems, Intelligent Transportation Systems (ITS), emerged as a result of these simulation technologies, which could be introduced and polished before being deployed. A smart city employs ICT strategies to address its sustainability issues. Smart cities can be used in a variety of areas, including architecture, transportation, health, and public safety. Intelligent Transportation Systems refers to the advances and study in the transportation field as a whole (ITS). The ITS is the result of a collaboration between information and communication technologies and urban transportation systems, which involve vehicles and networks that transport people and goods. Traffic network management has a significant impact on city growth. Because of the difficulty of traffic flow and its disorderly organization, researchers have had to rely on simulations rather than exact models (Singh et al., 2019). Drivers with successful traffic and road management plans may use simulations as intelligent perception devices. A smart device simulation's results may be used to model real-life scenarios. The SUMO simulator-based smart awareness framework for drivers allows for the generalization of an actual system, allowing drivers to focus on appealing events with the goal of reducing traffic congestion, reducing travel time, and desegregating various routes.

1.1. Motivation

Road traffic accidents are a serious socioeconomic problem, with no way of calculating the cost of human life, and they necessitate massive and ongoing government spending. To reduce the effects of accidents, various solutions have been proposed, one of which, Advanced Driver Assistance Systems.

In order to implement an effective solution, an intelligent approach is needed. The assistance platform is part of the vehicle, thus it must satisfy important requirements: first, provide the driver with real time information from road signs and Real time information is important for the safety of drivers, passengers, and pedestrians where the information provided warns the driver well in advance of any danger so that the appropriate manoeuvres can be made to correct the speed of the vehicle. Designing a new information system to improve traffic solutions is a very difficult task, especially when the system involves sociotechnical aspects, to which Simulation proved to be an effective approach to analysing and designing novel traffic solutions.

The vehicular safety application should be thoroughly tested before it is deployed in a real world to use. Simulator tool has been preferred over out door experiment because it simple, easy and cheap. In this thesis we will be focusing more on implementing assistance or awareness system for the drivers to help take the right decision in the right time using simulation softwares.

1.2 Research questions

To meet the goals of the expected solution, the following research questions will be investigated:

- Can the smart awareness system for drivers be a complement to other smart ITS?
- How can we incorporate and benefit from some emergent promising technologies, and more in the expected smart awareness system for drivers (SASD) solution?
- How can we embed intelligent mechanisms within the (SASD) solution for the ultimate goal of enabling drivers to make the right driving actions at the right time in the right place?

1.3 Objectives

There are several objectives from this master thesis which is :

- The first objective of this master thesis is to explore if the smart awareness system can extend the existing smart ITS with new functionality.
- The second objective is to include some emergent technologies within smart awareness system for drivers.
- The third objective is to set intelligent mechanisms within the (SASD) solution for the last goal of enabling drivers to make the right driving actions at the right time in the right place.

1.4 Thesis outline

- In Chapter 1, we present an introduction to the subject of work and contextualization, as well as the main goals and motivations for completing the thesis.
- In Chapter 2, literature review, introducing background concepts as well as some related works.
- In Chapter 3, proposed solution
- In Chapter 4, simulation
- In Chapter 5, Results
- In Chapter 6, Conclusion & Future work

Chapter 2

2. LITERATURE REVIEW

In this chapter, we look at some concepts that are important for a better understanding of this project, as well as a discussion of some related works that highlight their benefits and drawbacks.

2.1. Intelligent transportation system

Transportation is a necessary component of the activities of a city. Since it is the key infrastructure for people and goods trade, it has become an important part of every economy. Transportation networks are designed to transport people and other goods from one point to one or more destinations in an efficient and cost-effective manner. These systems have become increasingly complex and broad, resulting in their geographical and functional deployment (Rodrigue & Notteboom, 2019). The ITS idea was born in the mid-1980s when a group of transportation experts in the United States (US) realized the potential effect of the Information Age's computer and communications revolutions on urban transportation. The following modules, whose interactions are depicted in Figure 2.1, will form the base structure for ITS (Woodrow, 1998).

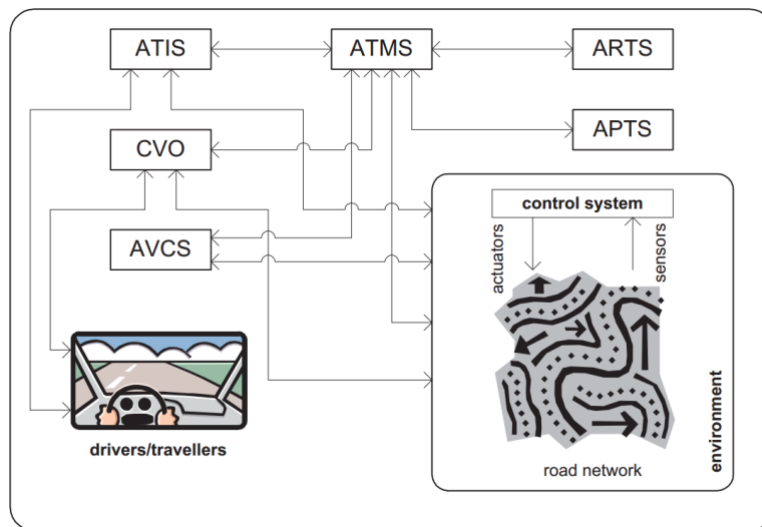


Figure 1: Intelligent Transportation System structure (Woodrow, 1998).

- **Advanced Traveler Information Systems (ATIS)** - provide a range of information for drivers, including navigation, path guidance, and danger warning, all tailored to the needs of the user.
- **Advanced Vehicle Control Systems (AVCS)** - these systems aid in driving tasks, especially in hazardous or unusual circumstances. The future vision of these devices is autonomous driving.

- **Commercial Vehicle Operations (CVO)** - provides facilities such as vehicle identification, loading monitoring, record keeping, and others to meet the unique needs of commercial transportation. This section aims to reduce the commercial sector's high costs.
- **Advanced Traffic Management Systems (ATMS)** - they play a critical role in ITS, monitoring, regulating, and handling traffic at all levels. Congestion can be reduced using methods such as automatic traffic signal timing, variable message signs (VMS), and virtual traffic lights. Communication is crucial in ITS, and it occurs between all individuals identified in the architecture of an Intelligent Transportation System. For the purposes of this study, we will only look at the ATIS subsystem from a holistic perspective of ITS related to traveler information.
- **Advanced Travellers Information Systems (smart awareness system)** - in intelligent transportation systems, Advanced Travelers Information Systems (ATIS) play a critical role. It provides pre trip and on the go travel information to enhance the trip's comfort, safety, and quality. In today's highly mobile society, accurate and timely traffic information will assist travelers in getting to their destinations quickly and safely. Advanced Traveler Information Systems (ATIS) were developed to meet this information requirement, allowing drivers to escape traffic and select more reliable and safer routes (Jamal, 2017).

2.2. Smart awareness system / Assistant driver system

In the mid-2000s, Fei-Yue Wang coined the term Artificial Transportation System or assistant driver system to describe the idea of Artificial Societies in the form of a generalized simulation paradigm of urban mobility systems (ATS). The meaning of ATS is a generalization of traffic simulation that combines the transportation system with other urban structures, such as logistic, social, and economic systems, to act as a coordinate method for transportation research, assessment, decision-making, and training (Li et al., 2007). Traditional traffic simulations are unable to capture the ambiguity that characterizes transportation networks due to their high complexity and complexities. Travelers can choose whether or not to fly, can change their plans at any time, and their decisions can be influenced by social, economic, or environmental factors. Due to the increased complexity of the test and validation task, which is made more difficult by real-time constraints and the existence of heterogeneous participating entities such as cars, urban and traffic infrastructures, and pedestrians, simulation is a key component in this new stage of mobility systems. Since it allows for resource allocation and multi-domain analysis, heterogeneity allows for a more complete and extended assessment or analysis in a transportation system. In short, understanding this definition is critical because we now have a tool in our hands that can help us understand how people make decisions and allow us to explore a wide range of circumstances. The traffic domain is no longer just an engineering problem; it has evolved into an interdisciplinary issue. In this context, sharing resources from various domains is critical, and one of the most recent trends is the Social Simulation sector. Advanced Driver Assistance Systems. As the name implies, these systems assist the driver by providing vital information about the traffic

environment or by acting in specific circumstances to protect the occupants of the vehicle or to facilitate driving.

In order to implement an effective solution, an intelligent approach is needed. The assistance platform is part of the vehicle, thus it must satisfy important requirements: first, provide the driver with real time information from road signs and Real time information is important for the safety of drivers, passengers, and pedestrians where the information provided warns the driver well in advance of any danger so that the appropriate manoeuvres can be made to correct the speed of the vehicle.

These exchanges will depend on several parameters, such as the ongoing situation (e.g., slippery road due to heavy rain or slow traffic flow), and road infrastructure (e.g., speed changes due to road works). To meet these goals, some emergent technologies that have proven their efficiency within the domain of transportation must be explored. Examples of such technologies include Machine Learning and Augmented Reality and some other simulation platform such as SUMO . Furthermore, the expected awareness system solution should ultimately motivate them to make the right driving actions at the right time in the right place for their benefits as well as for the benefits of all ITS stakeholders.

2.3. Simulation Frameworks

Simulators are useful instruments for evaluating vehicular network research. Typically, a vehicle mobility simulator determines vehicle motions, and the movement trace is then fed into a network simulator to mimic vehicle communication. As a result, both mobility and network simulators must be considered. We offer an overview of common vehicular networking simulators, as well as an experimental comparison of two prominent vehicular mobility simulators. Multiple types of simulators should be utilized in tandem to develop the proposed solution. The features of each simulator used in this master project will be highlighted in the sections that follow.

2.3.1. Mobility simulation

Simulation of Urban Mobility (SUMO) is a multi-modal traffic simulation tool that is open source, portable, tiny, and continuous. It is meant to manage enormous networks. SUMO is a collaborative effort between the German Aerospace Center and community members. Since 2001, it has been publicly available as open-source software (Hallerbach,2018). Software tools for simulation and

study of road traffic and traffic management systems are used in SUMO's traffic simulation. New traffic techniques can be tested in a simulation before being adopted in real-world scenarios. SUMO has also been presented as a component of a toolchain for developing and validating automated driving functionalities using various X-in-the-Loop and digital twin techniques. SUMO is utilized in research projects such as traffic forecasting, traffic signal evaluation, route selection, and vehicular communication systems. The open-source license allows SUMO users to experiment with new ideas by making changes to the program source code (Weber et al.,2020).

2.3.2. Network simulation

A network simulator is software that forecasts how a computer network will behave. Network simulators are utilized because communication networks have become too complex for standard analytical methods to provide a realistic knowledge of system behavior. The computer network is modeled in simulators with devices, links, applications, and so on, and the network performance is recorded. The most common technologies and networks in use today, such as 5G, the Internet of Things (IoT), Wireless LANs, mobile ad hoc networks, wireless sensor networks, and vehicular ad hoc networks, are all supported by simulators. Users can use network emulation to add actual devices and apps to a test network. Users can use network emulation to add real devices and applications to a simulated test network, which modifies packet flow to replicate the behavior of a live network. Live traffic can travel through the simulator and be influenced by the simulation's objects(Samaoui & Mansouri, 2015).

Typically, a network simulator is used to simulate computer networks. They are used to simulate VANETs by analyzing network protocol performance for node mobility and other relevant techniques. The majority of currently used network simulators were designed for MANETs and so require VANET additions (such as the usage of vehicle mobility generators) to mimic vehicular networks. Examples are OMNET++, NS-2 and NS-3 and OPNET.

1. NS2 and NS3

The NS-2 simulator was developed at the University of California as part of the Defense Advanced Research Projects Agency (DARPA)-funded Virtual Inter-Network Testbed (VINT) project. With the help of a wide group of users and developers, it has since been expanded and enhanced. The core kernel is written in C++, but the specifics of wired and wireless networks are handled by a variety of Tcl scripts (including some details of satellite and older technologies). The NS-2 simulation scenario scripts are written in TcL, which necessitates runtime compilation, slowing down the machine and increasing the cost of running a large-scale simulation. ((Pan et al., 2008).

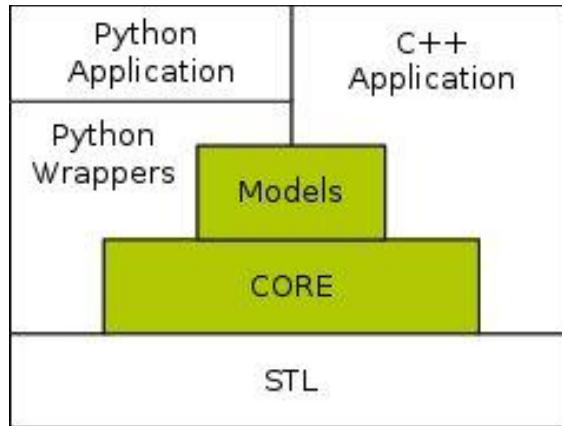


Figure 2 : Architecture of NS3 (Jawandhiya, et al., 2013)

2. OPNET

OPNET technology, Inc. created this simulator. OPNET was created at the Massachusetts Institute of Technology (MIT) and has been available as commercial software since 1987. It comes with a complete development framework for modeling communication networks and distributed systems. Discrete event simulations may be used to examine the action and efficiency of modeled structures. (Pan et al., 2008). OPNET's extensive features, according to the OPNET whitepaper, include:

1. A discrete event simulation engine that is fast.
2. A large component library with source code is available.
3. Object-oriented modeling is the third step.
4. Modeling environment with a hierarchical structure
5. Support for scalable wireless simulations
6. Graphical user interface (32-bit and 64-bit)
7. Wireless modeling that can be customized
8. Analytical, discrete event, and hybrid simulation
9. Kernel for 32-bit and 64-bit parallel simulation

3. OMNET++

OMNeT++ is a C++ simulation library and application that is mainly used to create network simulators. It is extensible, modular, and component-based. The term "network" refers to both wired and wireless communication networks, as well as on-chip networks, queueing networks, and other types of networks. Model systems, which are built as separate projects, provide domain-specific features such as support for sensor networks, wireless ad-hoc networks, Internet protocols, performance modeling, photonic networks, and so on. OMNeT++ includes an Eclipse-based integrated development environment, a graphical runtime environment, and a number of other features (Kabir et al, 2014). The models or modules of OMNeT++ are constructed from reusable components since OMNeT++ is designed to provide a component-based architecture. One of the key advantages of OMNeT++ is that modules are interchangeable and can be combined in a variety of ways (Pan, 2008).

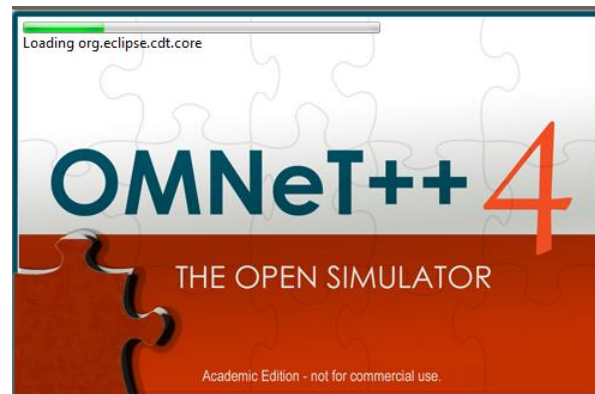


Figure 3 : :OMNET ++ Simulator

Components of OMNeT++ [OMNET] include:

1. A library of simulation kernels
2. NED topology description language compiler (nedc)
3. For NED files, a graphical network editor is available (GNED)
4. A graphical user interface (GUI) for running simulations, with links to the simulation executable (Tkenv)
5. A command-line user interface for running simulations (Cmdenv)
6. Vector plotting method with graphical output (Plove)
7. Visualization tool for graphical output scalars (Scalars)

2.3.3. VANETs simulation

The contact between cars has been studied in recent years as an important domain in computer and network science. The aforementioned communication could be accomplished using a Vehicular Ad-Hoc Network (VANET), which is similar to Mobile Ad-Hoc Networks (MANET) and is used for exchanging information between cars and roadside units. The objective and primary goal of VANETs is to provide comfort and protection to passengers in vehicles. To achieve the aforementioned target, an electronic device such as a wireless modem should be installed within cars to enable customers to communicate. Cars with ad-hoc network devices are called nodes in the VANET, and they can send and receive data with other cars and roadside units (RSUs) (Fiore et al ,2007).

VANET is a form of MANET that represents a specific research area (Mobile Ad- Hoc Networks). The VANET's core concept is straightforward and straightforward: develop a broad and inexpensive wireless technology to link vehicles to each other and to roadside units (RSU) for data transmission and reception. The following are some of the main characteristics of VANETs: In a VANET, cars and RSU are nodes. The nodes can move very quickly, and the network under consideration is highly dynamic, which means that the topology of the network is constantly changing as the nodes' positions and density change. In summary, VANET is a technology that allows vehicles to communicate with one another as well as with roadside units(Fiore et al ,2007).Wireless networking technology is one of the most effective solutions for dealing with traffic congestion, and it focuses on the third approach. Car to Car Communication (C2CC), also known as Vehicle to Infrastructure (V2I) and Vehicle to Vehicle (V2V) Communication, or V2X communication, is a form of wireless communication that aims to solve these issues. Intelligent Transportation Systems (ITSs) use V2V and V2I in a variety of ways to improve transportation efficiency. Because of its specific characteristics, the VANET topology is the most useful method for this wireless communication between vehicles. For example, the rapidly changing network topology and mobility induced by the high node travel speeds, as well as the constrained pattern due to the restricted roads, are some of the unique characteristics(Hartenstein & Laberteaux, 2010).

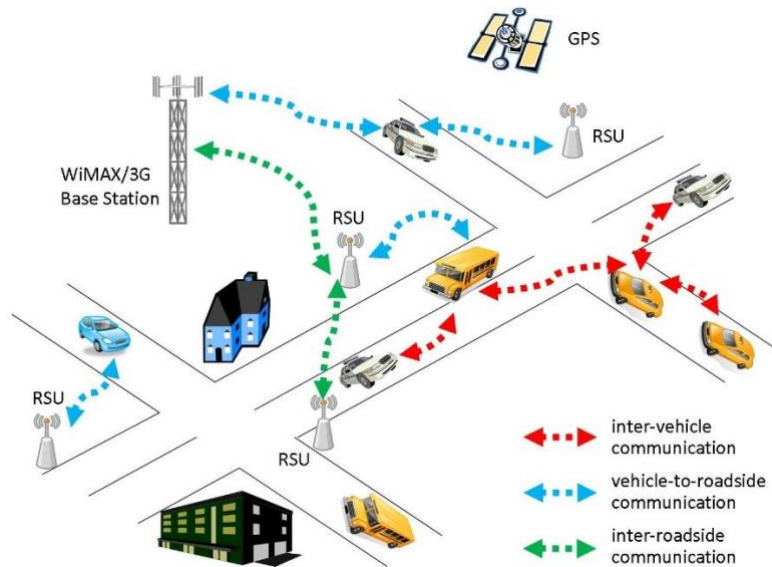


Figure 4: Type of communication in VANET(Hartenstein & Laberteaux, 2010).

2.4. Related studies

Al Sultan's work focuses on building a unique and nonintrusive driver behavior detection system on VANETs that uses a context-aware system to detect anomalous driving behaviors and notify other vehicles on the road to avoid accidents. A five-layer context-aware architecture is described, capable of gathering contextual information about the driving environment, reasoning about certain and uncertain contextual information, and reacting to it. The authors describe a real-time probabilistic model based on dynamic Bayesian networks (DBNs) that infers four categories of driving behavior (normal, drunk, reckless, and tiredness) by merging contextual information about the driver, the vehicle, and the surroundings. The dynamic behavior model can capture both the static and temporal components of the driver's activity, resulting in reliable and accurate behavior detection. The validity of our model and the necessity of integrating contextual information about the driver, the vehicle, and the surroundings are demonstrated by the evaluation of behavior detection using synthetic data (Al-Sultan et al, 2013).

Novel strategies for speed-based lane change, collision avoidance, and time of arrival (TOA) based localisation in Vehicular Ad Hoc Networks are proposed in this research (VANETs) by Agarwal. Because GPS requires a clear line of sight for reliable positioning and localization services, a Time of Arrival (ToA) based method has been developed for places where strong GPS signals are lacking. Other applications that we considered include collision avoidance employing automatic braking and camera-based surveillance. Simulations in Simulation of Urban Mobility (SUMO) and Network Simulator-2 were used to illustrate the algorithms' practicality and practicality (NS-2). Simulations in Simulation of Urban Mobility (SUMO) and Network Simulator-2 were used to

illustrate the algorithms' practicality and practicality (NS-2). To evaluate the effectiveness of the suggested system, we prototyped a functioning hardware and tested it on real automobiles. For smart, efficient, and remote traffic monitoring, a mobile app interface was built for the on-board unit. The proposed applications are built on the integrated VANET Cloud Computing architecture (Agarwal et al, 2018).

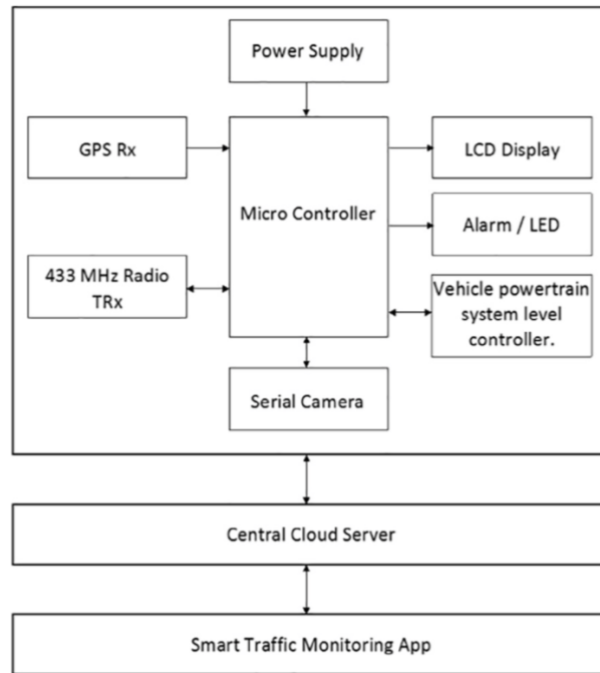


Figure 5 : Flow chart of the system (Agarwal et al, 2018)

Dmitry Kachan's work combines ns3 and MATLAB/Simulink in order to manage complex vehicle motion and wireless communication. To construct complex traffic movements, MATLAB is used, and ns3 is used to simulate wireless communication networks. To merge MATLAB/Simulink and ns3, socket programming is used. Simulink calculates future potential events at the end of each simulation stage, and MATLAB sends those instructions to ns3 through a socket connection. And until the simulation phase expires, ns3 executes those instructions. It sends the report to MATLAB at the end of the ns3 simulation process. Both simulators exchange messages containing status information for each node at each simulation stage. The interaction between ns3 and MATLAB is depicted in figure 6. This project primarily focuses on establishing a link between two simulators, but it does not describe any control strategies or evaluate the system using various vehicle movements or communication scenarios. However, we developed a system that incorporates several simulators, like mobility modeling, and we put it through its paces with various traffic scenarios (Kachan, 2010).

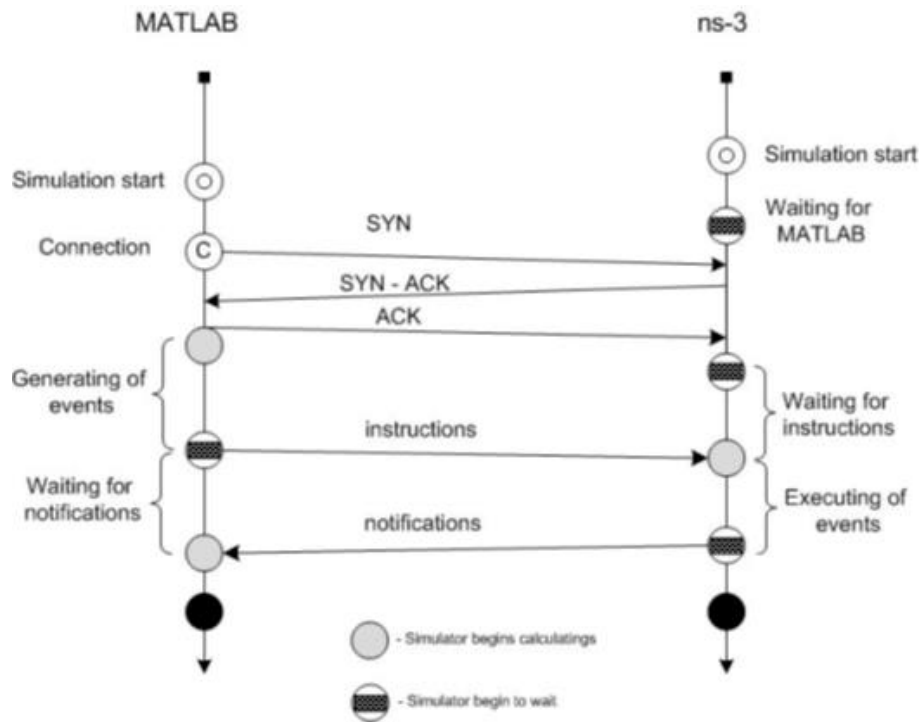


Figure 6 : Interaction between ns3 and MATLAB(Kachan, 2010)

IPG CarMaker and Simulation of Urban MObility (SUMO) were used by Jakob Kathes to integrate microscopic traffic simulation and vehicle dynamics simulation. As a mediator, MATLAB-Simulink is used to build the integrated structure. CarMaker for Simulink allows CarMaker to interact with Simulink, while TraCI4Matlab is used for traffic flow simulation. MATLAB/Simulink contact with SUMO is possible. SUMO is used to simulate all vehicles in the simulation, including the Ego car from CarMaker. The TraCI command "moveToVTD" is used to update the location of the Ego car in SUMO. During the simulation, the surrounding vehicles' IDs, locations, and speeds are retrieved and sent to CarMaker for updating within the CarMaker world. They put their framework to the test by using SUMO's high and low traffic volumes, as well as ACC's implementation of ADAS functionality to adapt the speed of the Ego Car in CarMaker. Their primary goal is to analyze vehicle dynamics in relation to traffic flow. Although we have created a framework that focuses on vehicular safety applications by detecting and preventing collisions using CACC and ACC, we have also created a framework that focuses on pedestrian safety applications. CarMaker has also been used for traffic and ADAS functions(Kathes et al., 2017).Chenxi Lei's research uses SUMO, Simulink, and OMNeT++ 2.18 to analyze string stability. String stability refers to the fact that any non-zero location, speed, or acceleration errors in a string of vehicles do not amplify as they spread upstream. Each simulator is dedicated to a single purpose. Simulink is in charge of vehicle behavior, including the ACC and CACC, SUMO of mobility, and OMNeT++ with MiXiM of contact networking behavior. The Real-Time Workshop tool in Simulink is used to transform the Simulink model into a C++ shared library,

which is then named in the SUMO source code to create a coupling between SUMO and Simulink. The bidirectional coupling between OMNeT++/MiXiM and SUMO is accomplished by sending TCP messages over the TRAFFIC Control Interface (TraCI). MiXiM collects data from all nodes (vehicles) at each simulation phase and sends it to SUMO via TraCI, where the acceleration data is fed into the Simulink model to calculate the next acceleration and velocity. Vehicles are transferred to SUMO as a result of this. The new trace is sent back to MiXiM, which adjusts the contact nodes based on the SUMO vehicle location data. They tested String Stability using the system by varying time headway, beacon rate, packet loss ratio, and ACC versus CACC. As previously mentioned, his main emphasis is on string stability and how it affects platooning applications. They also struggled to move between ACC and CACC-based controllers, while we were able to do so successfully. By considering the desired acceleration measured by them, we may switch between ACC and CACC controllers (Chenxi et al., 2012).

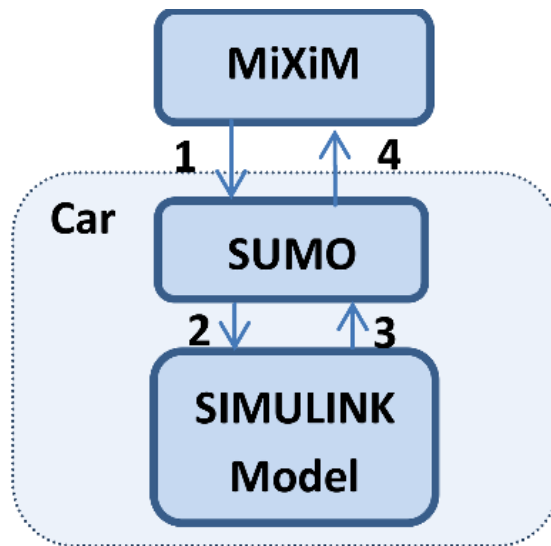


Figure 7: : Simulation Experimental Structure(Chenxi et al., 2012).

Amr Ibrahim and his colleagues developed a co-simulation system that included MATLAB for control algorithms, ns3 for network simulation, and SUMO for traffic behavior. This framework operates on two separate operating systems: MATLAB and SUMO run on Windows, and ns3 runs on a Linux virtual machine. The gui between ns3 and SUMO is MATLAB. TraCI is used to create a TCP link between SUMO and MATLAB (Traffic Control Interface). For the TCP link between MATLAB and ns3, socket programming is used. Figure 7 depicts the co-simulation framework.

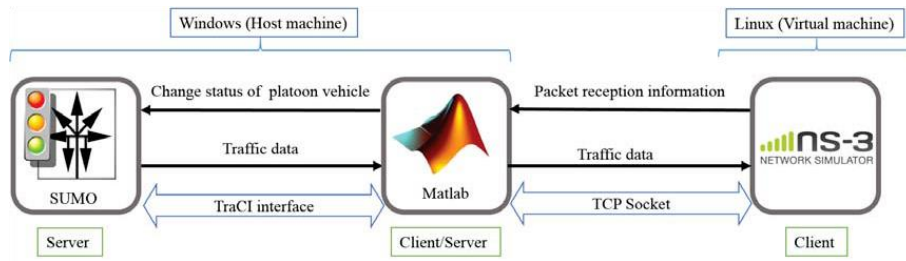


Figure 8 : ns3 and SUMO Interaction through MATLAB

Chapter 3

3. METHODOLOGY

Fundamentals of the proposed solution

In reality, using simulation to solve traffic issues is a more realistic and intelligent approach (Wilkie,2015). However, there is no proper simulation process for drivers' decision-making in an advanced travelers information system. This chapter summarizes the project's key criteria and describes the proposed solution and design. To create the desired structure, we'll need a community of individuals as well as an atmosphere in which their interactions can take place.

When dealing with atomic entities in the transportation domain, where the resolution is down to the vehicle or to the drivers, a smart awareness system or traveler information system seems to be the best way to reflect the road traffic environment and the driver entities that reside and communicate in it. A traffic simulator that implements the smart warning system paradigm must have a good API for accessing the vehicles and other traffic variables so that we can build, manage, and track their states from an external application.

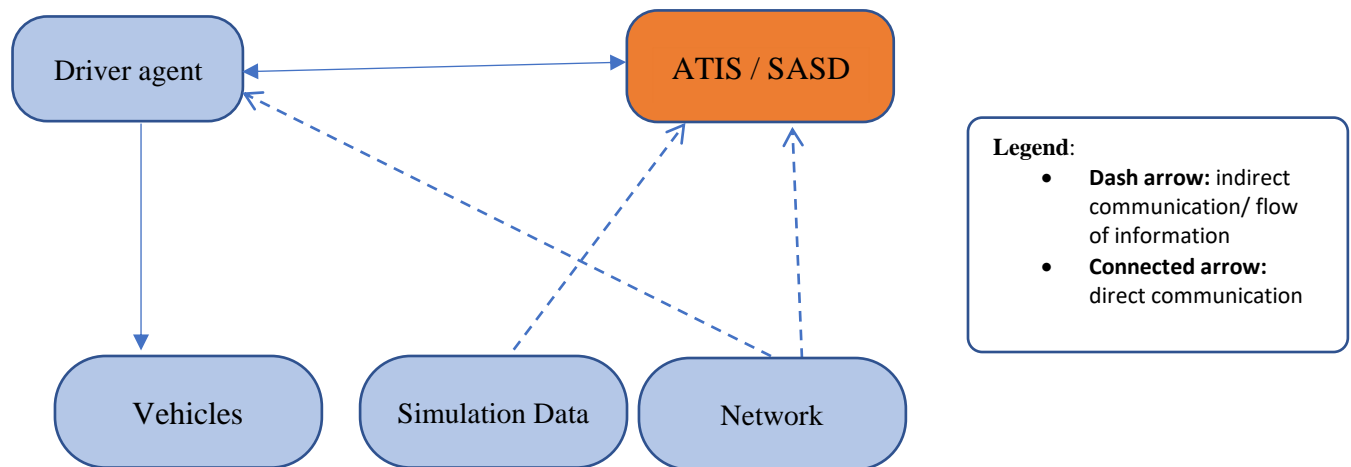


Figure 9 : Basic framework setup

Figure 9 shows the general view of the proposed solution. The driver agents should be able to communicate with one another, request information, and share it. The ATIS or SASD can use its network and simulation data interpretation to gather information. The drivers would then view the surrounding network and make driving decisions. From this master thesis my focus will be on SASD which have different components like, data processing, traffic simulation and traffic

recommendation. Simulation data will help us here to gather all the necessary information needed by the driver and then network simulators will be used to work with VANET simulator in order to have connect between cars and surrounding area in the road like traffic lights. Below we will discuss the importance and the purpose of each simulators used.

Mobility Simulators

SUMO

(Simulation of Urban Mobility) is a very efficient (microscopic) traffic simulator that is also open source. In a wide range of VANET ventures, SUMO has been used. The road network, vehicle forms, and vehicle routes are all highly customizable, allowing for unique simulations. Furthermore, SUMO can interact bi-directionally with any network simulator that implements TraCI thanks to the Traffic Control Interface (TraCI). This enables the traffic simulator's findings to influence the network simulator and vice versa (Krajzewicz, 2010).

Network simulators

Network Simulator (NS) is a discrete event-driven network simulation tool for understanding communication networks' dynamic nature. Network Simulator 2 (NS2) is a free and open source widely used tool that allows for the simulation of many protocols over wired and wireless networks. It offers a highly modular platform for wired and wireless simulations, with support for a variety of network elements, protocols, traffic, and routing types (Samaoui, 2015).

VANET simulators

VANET simulators can simulate traffic as well as networks, or they can combine the two. Veins, TraNS, MOVE, NCTUns, GrooveNet, and MobiREAL are some examples. The simulation model for V2X systems must take into account a number of factors. One of the most significant is the response of the driver to novel V2X applications. When a driver receives crash warning signals, he will either leave the street or touch the brakes, depending on the distance between the driver and the incidents, as well as whether or not there are any exits available. The VANET simulator must be able to alter the behavior of the vehicles in response to given application contexts. The VANET simulator, also known as an integrated system, is a form of simulation (Sommer et al, 2011).

There are multiple types of simulators that should work together. We will highlight in what follows the features of each simulator used to implement this master project. In order to implement this project in efficient way we follow several steps starting from literature review, in this step we explored the related works with their main objectives and strength and. Moreover, we describe the different type of simulation starting from network simulation to mobility simulation and VANET simulation. Second step is methodology, in this step we describe the proposed solution that we want to implement in this master project, then we talk about the different type of simulators tools that we will use in order to implement this project. In the following step we also identify rooms for improvement in addition, we analyze the necessary requirements to implement our solution for example we start by the solution design and setup then simulation and results and at the end we close the project by recommendation and future work.

Chapter 4

4. SIMULATION SETUPS & SCENARIOS

4.1. Part 1: Software setup

This segment contains a practical traffic and map. The Network simulator configuration is then described, followed by the simulation scenario. In the following section, the findings will be discussed. Scenarios in the SUMO simulator are divided into two parts: the road network (maps), which includes highways, streets, traffic lights, junctions, and so on, and the traffic demand, which includes information about the cars driving, such as their speed, some physical assets, location, departure and arrival times and locations, and so on. The SUMO Road network can be created using the “netgen” application included in the SUMO kit, or by importing a digital road map. Different sources may be used to describe traffic demand. In this case we will create our own network in SUMO using Nodes and Edges for the first scenarios and for other scenarios we will import map from open street map.

Three steps must be taken to set up a SUMO simulation.

The road networks that will be used to simulate vehicle traffic are needed. Three approaches may be used to achieve these objectives:

1. Creating an abstract road network, b. creating an own XML definition and importing it with the NETCONVERT tool, and c. importing an existing road network with the NETCONVERT tool.
2. Each vehicle should determine its routes, which are a list of all edges that the vehicle must pass, as well as the edges themselves.
3. This can be accomplished by:
 - a. defining explicit routes on the road network.
 - b. using predefined routes and only enabling a portion of them.
 - c. Making up random routes
 - d. Using existing routes as a starting point. It will also be necessary to compute the dynamic user assignment and calibrate the simulation using the provided measures if necessary.
4. 3- Finally, you must run the simulation.

To create simple network

File used to create SUMO road map:

- Node file (myown.nod.xml file)
- Edge file (myown.edge.xml file)
- Network file (myown.net.xml file)
- Route file (myown.row.xml file)
- Configuration file ((myown.sumocfg file)

Node

All nodes have a unique position (the x- and y-coordinates, which define the distance to the origin in meters) as well as a unique identifier for future reference. As a result, the node file looks like this:

```
1 ▼ <nodes> <!-- The opening tag -->
2
3   <node id="l1" x="0.0" y="1000.0" type="priority"/>
4   <node id="l2" x="0.0" y="600.0" type="priority"/>
5   <node id="l3" x="0.0" y="300.0" type="traffic_light"/>
6   <node id="l4" x="0.0" y="0.0" type="priority"/>
7
8   <node id="m1" x="500.0" y="1000.0" type="traffic_light"/>
9   <node id="m2" x="500.0" y="600.0" type="traffic_light"/>
10  <node id="m3" x="500.0" y="300.0" type="priority"/>
11  <node id="m4" x="500.0" y="000.0" type="priority"/>
12
13  <node id="r1" x="1000.0" y="1000.0" type="priority"/>
14  <node id="r2" x="1000.0" y="600.0" type="priority"/>
15  <node id="r3" x="1000.0" y="300.0" type="priority"/>
16  <node id="r4" x="1000.0" y="0.0" type="traffic_light"/>
17
18 </nodes> <!-- The closing tag -->|
19
```

Figure 10 :Node file

Edges

We must now link the nodes using edges. For future reference, we have a target node id, a source node id, and an edge id. Since edges are guided, any vehicle traveling along them will begin at the node given information and end at the node given into. As result , the edges file looks like this :

```
1 <edges>
2 <edge id="rl1m1" from="l1" to="m1" priority="2" numLanes="1" speed="11.11"/>
3 <edge id="ll1m1" from="m1" to="l1" priority="3" numLanes="1" speed="13.89"/>
4 <edge id="rl2m2" from="l2" to="m2" priority="1" numLanes="1" speed="11.11"/>
5 <edge id="ll2m2" from="m2" to="l2" priority="2" numLanes="1" speed="11.11"/>
6 <edge id="rl3m3" from="l3" to="m3" priority="3" numLanes="1" speed="13.89"/>
7 <edge id="ll3m3" from="m3" to="l3" priority="1" numLanes="1" speed="11.11"/>
8 <edge id="rl4m4" from="l4" to="m4" priority="2" numLanes="1" speed="11.11"/>
9 <edge id="ll4m4" from="m4" to="l4" priority="3" numLanes="1" speed="13.89"/>
10
11 <edge id="rm1r1" from="m1" to="r1" priority="2" numLanes="1" speed="11.11"/>
12 <edge id="lm1r1" from="r1" to="m1" priority="3" numLanes="1" speed="13.89"/>
13 <edge id="rm2r2" from="m2" to="r2" priority="1" numLanes="1" speed="11.11"/>
14 <edge id="lm2r2" from="r2" to="m2" priority="2" numLanes="1" speed="11.11"/>
15 <lane index="0" disallow="all"/> </edge>
16 <edge id="rm3r3" from="m3" to="r3" priority="3" numLanes="1" speed="13.89"/>
17 <edge id="lm3r3" from="r3" to="m3" priority="1" numLanes="1" speed="11.11"/>
18 <edge id="rm4r4" from="m4" to="r4" priority="2" numLanes="1" speed="11.11"/>
19 <edge id="lm4r4" from="r4" to="m4" priority="3" numLanes="1" speed="13.89"/>
20
21 <edge id="dl1l2" from="l1" to="l2" priority="2" numLanes="1" speed="11.11"/>
22 <edge id="ul1l2" from="l2" to="l1" priority="3" numLanes="1" speed="13.89"/>
23 <edge id="dl2l3" from="l2" to="l3" priority="1" numLanes="1" speed="11.11"/>
24 <edge id="ul2l3" from="l3" to="l2" priority="2" numLanes="1" speed="11.11"/>
25 <edge id="dl3l4" from="l3" to="l4" priority="3" numLanes="1" speed="13.89"/>
26 <edge id="ul3l4" from="l4" to="l3" priority="1" numLanes="1" speed="11.11"/>
27
28
29 <edge id="dm1m2" from="m1" to="m2" priority="2" numLanes="1" speed="11.11"/>
30 <edge id="um1m2" from="m2" to="m1" priority="3" numLanes="1" speed="13.89"/>
31 <edge id="dm2m3" from="m2" to="m3" priority="1" numLanes="1" speed="11.11"/>
32 <edge id="um2m3" from="m3" to="m2" priority="2" numLanes="1" speed="11.11"/>
33 <edge id="dm3m4" from="m3" to="m4" priority="3" numLanes="1" speed="13.89"/>
34 <edge id="um3m4" from="m4" to="m3" priority="1" numLanes="1" speed="11.11"/>
35
36 <edge id="dr1r2" from="r1" to="r2" priority="2" numLanes="1" speed="11.11"/>
37 <edge id="ur1r2" from="r2" to="r1" priority="3" numLanes="1" speed="13.89"/>
38 <edge id="dr2r3" from="r2" to="r3" priority="1" numLanes="1" speed="11.11"/>
39 <edge id="ur2r3" from="r3" to="r2" priority="2" numLanes="1" speed="11.11"/>
40 <edge id="dr3r4" from="r3" to="r4" priority="3" numLanes="1" speed="13.89"/>
41 <edge id="ur3r4" from="r4" to="r3" priority="1" numLanes="1" speed="11.11"/>
42
43 </edges>
44
```

Figure 11 : Edges file

A network file defines SUMO road networks in more detail. Lanes are described as edges in a directed graph with vertices representing connections between lanes in the network file. Each lane has its own set of characteristics, such as speed limits and turning restrictions. Lane connections may be as simple as a direction shift or as complicated as multilane intersections with traffic lights or priority traffic direction. Although the program is very complex, it does come with a set of tools for creating SUMO road networks. The NETGEN utility can be used to build simple geometric road networks. The NETCONVERT utility can be used to import map data from a variety of sources to model real-world road networks.

This information should be saved in a file called myown.edg.xml. We can use the first SUMO tool to build a network now that we have nodes and edges. Make sure **NETCONVERT** is in the PATH and then call it.

- **To call NETCONVERT**

```
$ netconvert --node-files=myown.nod.xml --edge-files=myown.edg.xml  
file=myown.net.xml
```

Once you get a success message, the conversion is successful and it creates a file called **myown.net.xml** file which is our network .

Routes

We still need a car now that we have a network file. Vehicles in SUMO are classified according to their basic characteristics, such as weight, acceleration and deceleration, and maximum speed. It also needs a so-called sigma parameter, which adds some random behavior as a result of the car-following model used. Setting it to 0 indicates that the car is deterministic.

Now we define a route for the vehicle, which consists solely of the two previously described edges. The need for two edges arises from the fact that in SUMO, the car vanishes as soon as it reaches the last edge of its own path (the position of a car is defined by the position of its front). To create a route file, we need to use **randomTrips.py file** and then also to run the simulation we need Radom trips.

- To create Random trips

```
$ python /home/meamrashid/sumo/tools/randomTrips.py -n myown.net.xml -r myown.r  
100 -l
```

Success message indicates its successful and it generated three files which is:

- trips.trip.xml file
- rou.xml file
- rou.alt.xml file.

```

<routes xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://sumo.dlr.de/xsd/routes_file.xsd">
  <vehicle id="0" depart="0.00">
    <route edges="rm1r1 dr1r2 ur1r2 lm1r1 ll1m1 dl1l2"/>
  </vehicle>
  <vehicle id="1" depart="1.00">
    <route edges="ll2m2 rl2m2 um1m2 ll1m1"/>
  </vehicle>
  <vehicle id="2" depart="2.00">
    <route edges="rm3r3 lm3r3 um2m3 rm2r2"/>
  </vehicle>
  <vehicle id="3" depart="3.00">
    <route edges="um2m3 um1m2 dm1m2"/>
  </vehicle>
  <vehicle id="4" depart="4.00">
    <route edges="lm3r3 ll3m3 ul2l3 dl2l3"/>
  </vehicle>
  <vehicle id="5" depart="5.00">
    <route edges="dr3r4 lm4r4 ll4m4 ul3l4 ul2l3 ul1l2 rl1m1"/>
  </vehicle>
  <vehicle id="6" depart="6.00">
    <route edges="dl1l2 dl2l3 rl3m3 rm3r3 dr3r4"/>
  </vehicle>
  <vehicle id="7" depart="7.00">
    <route edges="rm1r1 dr1r2 dr2r3 lm3r3 ll3m3 ul2l3 rl2m2"/>
  </vehicle>
  <vehicle id="8" depart="8.00">
    <route edges="dr3r4 lm4r4 ll4m4 ul3l4 rl3m3"/>
  </vehicle>
  <vehicle id="9" depart="9.00">
    <route edges="ul2l3 rl2m2 um1m2 ll1m1 dl1l2"/>
  </vehicle>
  <vehicle id="10" depart="10.00">
    <route edges="ul1l2 rl1m1 dm1m2 rm2r2"/>
  </vehicle>
  <vehicle id="11" depart="11.00">
    <route edges="ll4m4 ul3l4 rl3m3 um2m3 um1m2 rm1r1 dr1r2"/>
  </vehicle>
  <vehicle id="12" depart="12.00">
    <route edges="ur2r3 dr2r3 dr3r4 lm4r4 ll4m4"/>
  </vehicle>
  <vehicle id="13" depart="13.00">
    <route edges="dm1m2"/>
  </vehicle>
  <vehicle id="14" depart="14.00">
    <route edges="ul3l4 dl3l4 rl4m4"/>
  </vehicle>
</routes>

```

Figure 12:Routes file

Configuration

To run the simulation, we need to create configuration file and to do this:

```

<configuration>
  <input>
    <net-file value="myown.net.xml"/>
  </input>
  <route-files value="myown.rou.xml"/>
  <additional-files value="myown.poly.xml"/>
</configuration>
<time>
  <begin value="0"/>
  <end value="100"/>
  <step-length value="0.1"/>
</time>

```

Summary file

To be able to have one file with all needed information that you might need it in the analysis of your case like (speed, collision , waiting time) you create this method by:


```
$ sumo -c myown.sumocfg --summary myown_summary.xml
```

Sumo-GUI file

Sumo-gui allows the selection of arbitrary network elements such as edges, lanes, and junctions, which can then be saved to (and loaded from) a file for further processing. SUMO GUI file. To run the simulation in a GUI:

```
$ sumo-gui -c myown.sumocfg
```

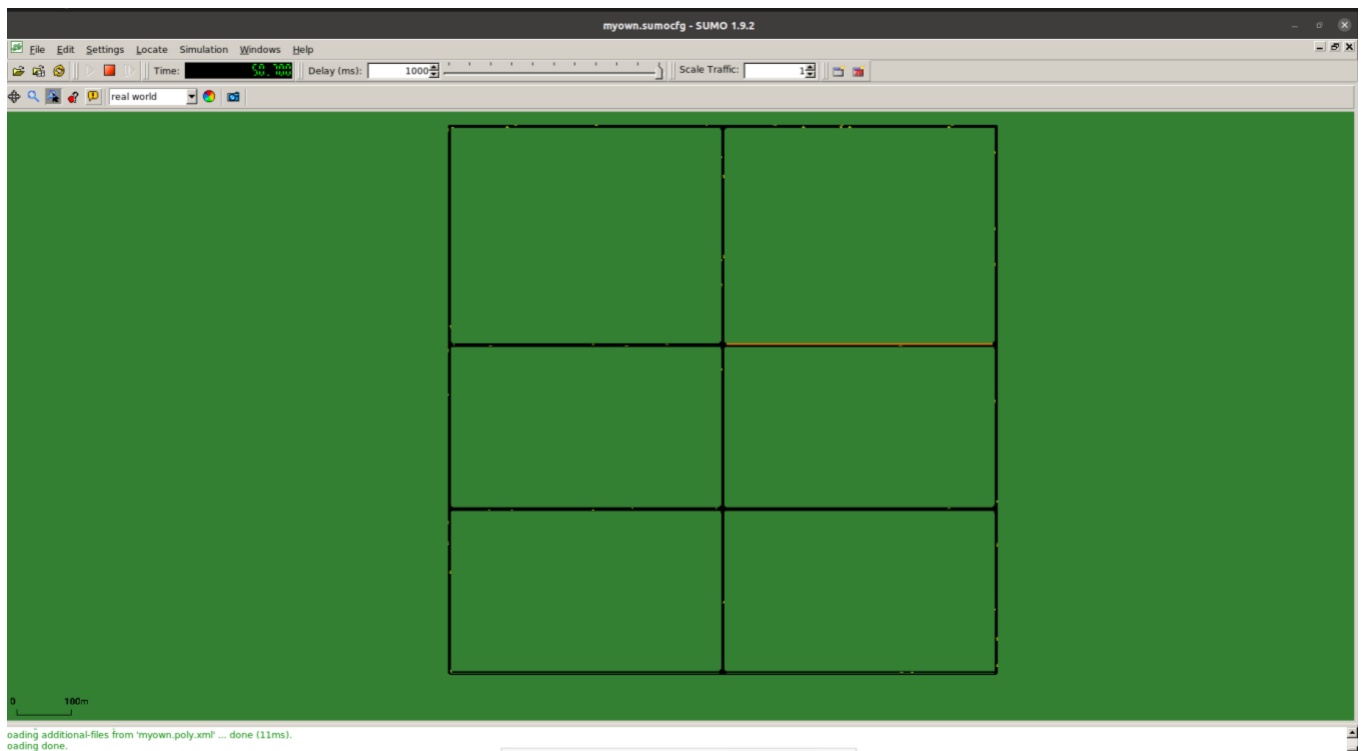


Figure 13: Street Map

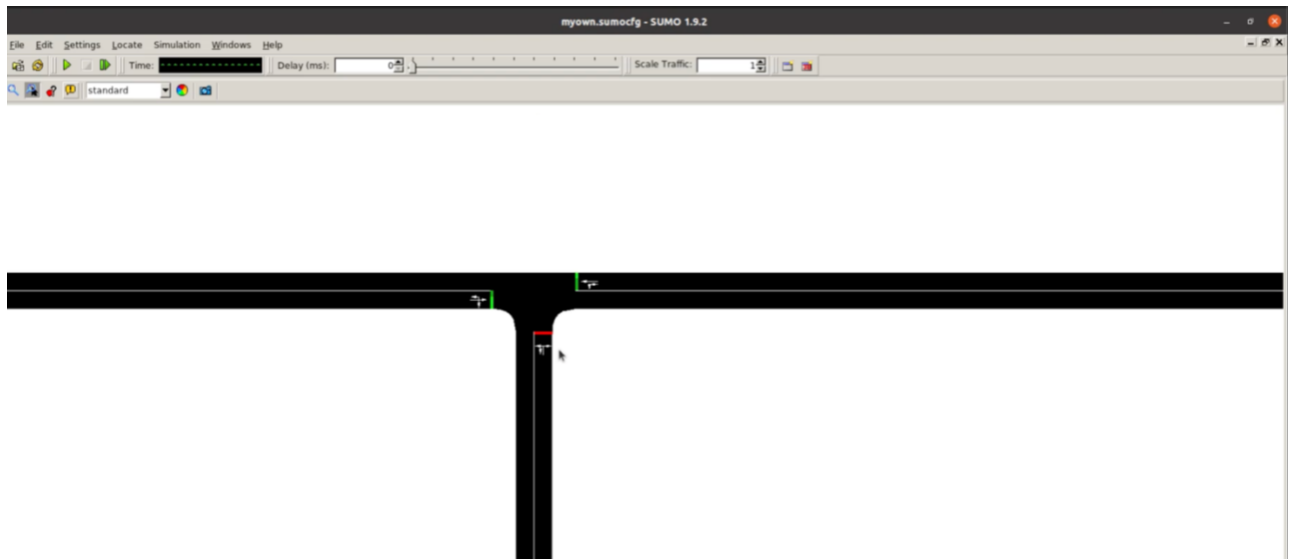


Figure 14 : Traffic lights

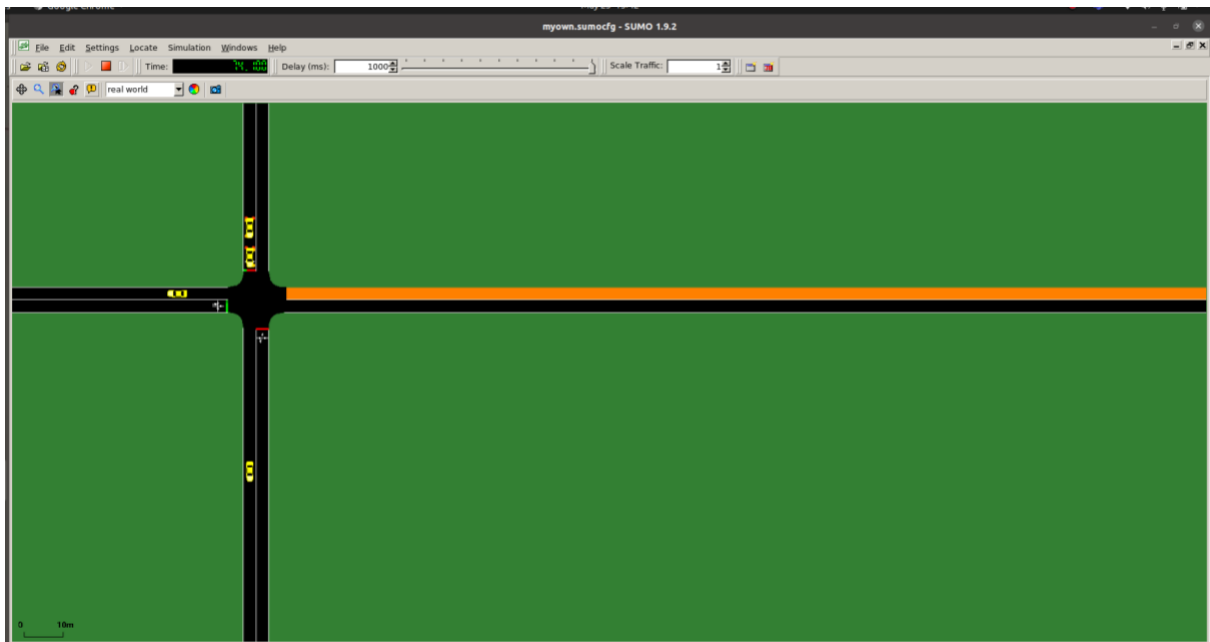


Figure 15: Block road

Once we have the summary file that we created before we can calculate different type of data like all the data related to vehicle itself and the road as well.

CSV file

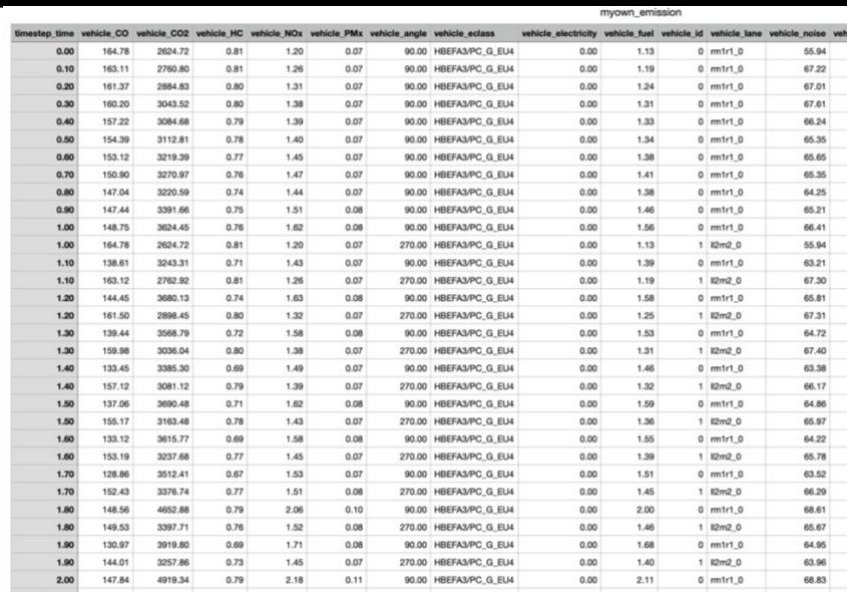
To convert file from xml format to csv , CSV(Comma Separated Values) will enable us to plotting graphs and other results. To do this we use the option from SUMO `xml2csv.py` (that converts xml file to csv) using the following commands:

```
$ cd /home/pradeepkumar/sumo/tools/xml
$ python xml2csv.py /home/meamrashid/myownsumomap/myown_statistics.xml
```

Emission & Collision

To get emission, collision and statistics file in the sumo command, use the following command.

```
$ sumo -c myown.sumocfg --summary summary.xml --device.battery.track-fuel true --collision-output myown_collision.xml --statistic-output myown_statistics.xml --emission-output myown_emission.xml
```



myown_emission														
timestep	time	vehicle_CO	vehicle_CO2	vehicle_HC	vehicle_NOx	vehicle_PMx	vehicle_angle	vehicle_ectassa	vehicle_electricity	vehicle_fuel	vehicle_id	vehicle_lane	vehicle_noise	vehic
0.00	164.78	2624.72	0.81	1.20	0.07	90.00	HBEFAS/PC_G_EU4	0.00	1.13	0	mm1r_0	55.94		
0.10	163.11	2760.80	0.81	1.26	0.07	90.00	HBEFAS/PC_G_EU4	0.00	1.19	0	mm1r_0	67.22		
0.20	161.37	2884.83	0.80	1.31	0.07	90.00	HBEFAS/PC_G_EU4	0.00	1.24	0	mm1r_0	67.01		
0.30	160.20	3043.52	0.80	1.38	0.07	90.00	HBEFAS/PC_G_EU4	0.00	1.31	0	mm1r_0	67.61		
0.40	157.22	3084.68	0.79	1.39	0.07	90.00	HBEFAS/PC_G_EU4	0.00	1.33	0	mm1r_0	66.24		
0.50	154.39	3112.81	0.78	1.40	0.07	90.00	HBEFAS/PC_G_EU4	0.00	1.34	0	mm1r_0	65.35		
0.60	153.12	3219.39	0.77	1.45	0.07	90.00	HBEFAS/PC_G_EU4	0.00	1.38	0	mm1r_0	65.65		
0.70	150.90	3270.97	0.76	1.47	0.07	90.00	HBEFAS/PC_G_EU4	0.00	1.41	0	mm1r_0	65.35		
0.80	147.04	3220.59	0.74	1.44	0.07	90.00	HBEFAS/PC_G_EU4	0.00	1.38	0	mm1r_0	64.25		
0.90	147.44	3391.66	0.75	1.51	0.08	90.00	HBEFAS/PC_G_EU4	0.00	1.46	0	mm1r_0	65.21		
1.00	148.75	3624.45	0.76	1.62	0.08	90.00	HBEFAS/PC_G_EU4	0.00	1.56	0	mm1r_0	66.41		
1.00	164.78	2624.72	0.81	1.20	0.07	270.00	HBEFAS/PC_G_EU4	0.00	1.13	1	k2m2_0	55.94		
1.10	138.61	3243.31	0.71	1.43	0.07	90.00	HBEFAS/PC_G_EU4	0.00	1.29	0	mm1r_0	63.21		
1.10	163.12	2762.92	0.81	1.26	0.07	270.00	HBEFAS/PC_G_EU4	0.00	1.19	1	k2m2_0	67.30		
1.20	144.45	3680.13	0.74	1.63	0.08	90.00	HBEFAS/PC_G_EU4	0.00	1.58	0	mm1r_0	65.81		
1.20	161.50	2898.45	0.80	1.32	0.07	270.00	HBEFAS/PC_G_EU4	0.00	1.25	1	k2m2_0	67.31		
1.30	139.44	3568.79	0.72	1.58	0.08	90.00	HBEFAS/PC_G_EU4	0.00	1.53	0	mm1r_0	64.72		
1.30	159.98	3036.04	0.80	1.38	0.07	270.00	HBEFAS/PC_G_EU4	0.00	1.31	1	k2m2_0	67.40		
1.40	133.45	3385.30	0.69	1.49	0.07	90.00	HBEFAS/PC_G_EU4	0.00	1.46	0	mm1r_0	63.38		
1.40	157.12	3081.12	0.79	1.39	0.07	270.00	HBEFAS/PC_G_EU4	0.00	1.32	1	k2m2_0	66.17		
1.50	137.06	3690.48	0.71	1.62	0.08	90.00	HBEFAS/PC_G_EU4	0.00	1.59	0	mm1r_0	64.86		
1.50	155.17	3163.48	0.78	1.43	0.07	270.00	HBEFAS/PC_G_EU4	0.00	1.36	1	k2m2_0	65.97		
1.60	133.12	3615.77	0.69	1.58	0.08	90.00	HBEFAS/PC_G_EU4	0.00	1.55	0	mm1r_0	64.22		
1.60	153.19	3237.68	0.77	1.45	0.07	270.00	HBEFAS/PC_G_EU4	0.00	1.39	1	k2m2_0	65.76		
1.70	128.86	3512.41	0.67	1.53	0.07	90.00	HBEFAS/PC_G_EU4	0.00	1.51	0	mm1r_0	63.52		
1.70	152.43	3376.74	0.77	1.51	0.08	270.00	HBEFAS/PC_G_EU4	0.00	1.45	1	k2m2_0	66.29		
1.80	148.56	4652.88	0.79	2.06	0.10	90.00	HBEFAS/PC_G_EU4	0.00	2.00	0	mm1r_0	68.61		
1.80	149.53	3397.71	0.76	1.52	0.08	270.00	HBEFAS/PC_G_EU4	0.00	1.46	1	k2m2_0	65.67		
1.90	130.97	3919.80	0.69	1.71	0.08	90.00	HBEFAS/PC_G_EU4	0.00	1.68	0	mm1r_0	64.95		
1.90	144.01	3257.86	0.73	1.45	0.07	270.00	HBEFAS/PC_G_EU4	0.00	1.40	1	k2m2_0	63.96		
2.00	147.84	4919.34	0.79	2.18	0.11	90.00	HBEFAS/PC_G_EU4	0.00	2.11	0	mm1r_0	68.83		

Figure 16:Emission

Floating car data (FCD)

To get all the data about the vehicle like (speed , travel time , lane changing)

```
$ sumo -c myown.sumocfg --fcd-output myown.sumo.xml
```

Simulation of a Network

In SUMO, the previous section prepares the practical traffic and map. The network simulator must then be configured in order to run the VANET simulation. Cologne's practical traffic is modelled as a VANET, with each car communicating with the others. The simulation scenario is run for 360 seconds using the setup, dataset, and parameters. The time it takes to simulate a VANET on a large scale is one of the challenges. OMNET++, for example, spends more than 30 hours in real time for a 360-second simulation. To implement our scenarios and connect the vehicles we have to use VANET then we have to use **TraceExporter** to export our xml files to ns3 or ns2 or OMNET++. This command is used to create our xml files to TCL (Tool Command Language) Files that can be run under ns2 .

```
$ python /home/meamrashid/sumo/tools/traceExporter.py --fcd-input myown.sumo.xml --ns2config-output myown.tcl --ns2activity-output myown_activity.tcl --ns2mobility-output myown_mobility.tcl
```

4.2. Part 2 : Open street map (OSM)

After we build our own network and we used different commands to collect different type of data to make the solution provided more efficient using VANET. Now in this part we will collect same type of data like data related to vehicles itself and the road as well but with real life example using open street map. For road network and traffic demand simulation, a data set from is used. The realistic traffic demand for the Belgian city of Hasselt has been presented, along with a realistic map of the city. The automobiles' journey the city's street network is also imported from the OpenStreetMap (OSM) database. This information pertains to a region of approximately It covers an area of 102.24 km and is home to about 700.000 vehicles. Both of these datasets have been uploaded to SUMO. The Hasselt map in SUMO is shown in Figure 9.

```
$ cd sumo/tools/  
$ python osmWebWizard.py  
It will open a browser with a OSM (Map)
```

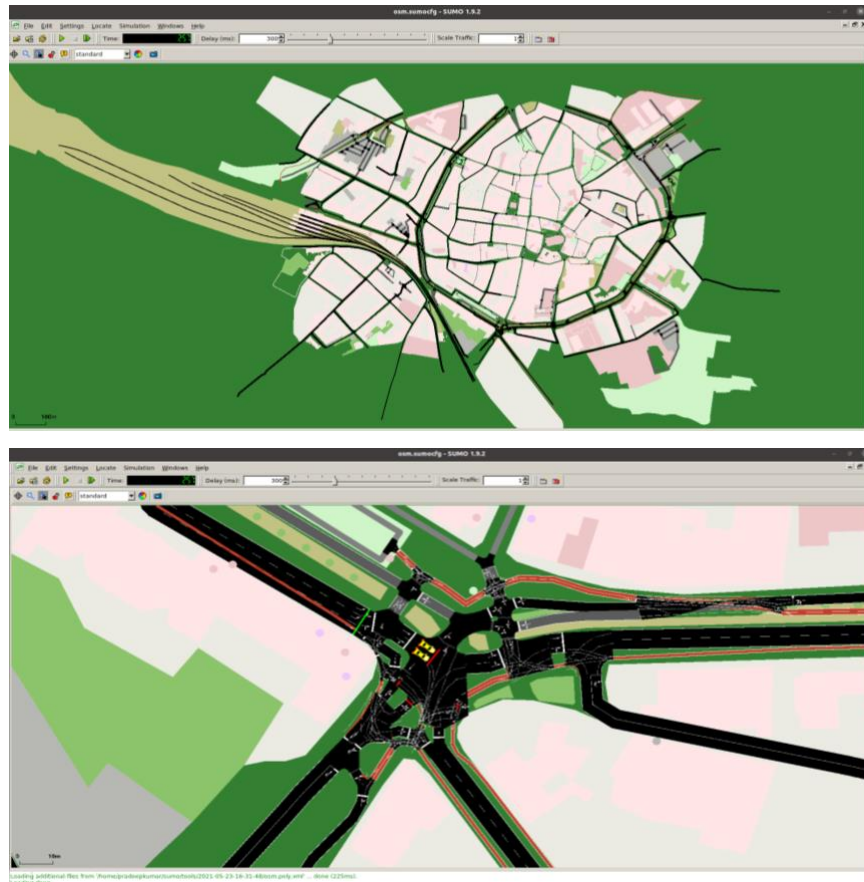


Figure 17: Hasselt city Map and simulation in SUMO

We used the same command that is mentioned up to collect all data needed like the emissions data and collisions etc. with VANET as well.

Chapter 5

5. Scenario results

VANET and SUMO are used in the simulation. The cars travel through Hasselt and communicate with neighboring cars through broadcasting, collecting the necessary data. The main aim of this chapter is to look at the outcomes and make comparisons using various scenarios in a real-world setting. As a result, multiple reviews of simulation data are carried out. One of the most critical V2I applications is communication between cars and traffic lights, which allows for dynamic and automatic traffic lights, which can provide many benefits such as reducing traffic congestion, reducing fuel consumption and pollution, and so on. Additionally, for each car, several statistics about traffic simulation are generated, such as travel time, waiting time, emissions, and fuel consumption; or the total amount of car emissions in the street during the simulation, fuel consumption, number of vehicles, and so on; or the total amount of car emissions in the street during the simulation, for each street. This chapter describes the software using a realistic map of Hasselt city, Belgium to produce a realistic and reliable result.

Table 1: SUMO statics simulation

Vehicle	Street
Speed	Maximum speed of cars
Vehicle route and travel time	Travel time on the street
Vehicle fuel consumption	Amount of fuel consumption
Gas (CO ₂ , CO,HC)	Amount of gas by vehicle on the road
Noise	Average noise by car on the street
Waiting time & time loss	Waiting time and time spend in case of there is accident or traffics

There are 5 scenarios were taken based on the number of vehicles namely

- Passenger Cars
- Trucks
- Motor Bikes

Table 2 : Various parameters of simulation

Name of the Parameter	Value
Number of Vehicles	Scenario 1 (166 vehicles) Scenario 2 (238 Vehicles) Scenario 3 (317 Vehicles) Scenario 4 (1543 Vehicles) Scenario 5 (1833 Vehicles)
Collision	No collision, Teleports, Warn and Remove Be default
Teleports	Jam, Wrong Lane and Yield
Location	Hasselt
VANETs Usage	With and Without VANETs

We have plotted the following three characteristics based on the results obtained from the SUMO and NS2 Simulation. Figure 18 shows the Waiting time of vehicles and the Time loss in the SUMO simulation. There x axis refers the increasing number of vehicles, and the y axis refers the waiting time of the vehicles in the traffic signals and other places during collision. For smart driving system, the waiting time and time loss (the loss incurred by the vehicles before reaching their destination) is of prime importance.

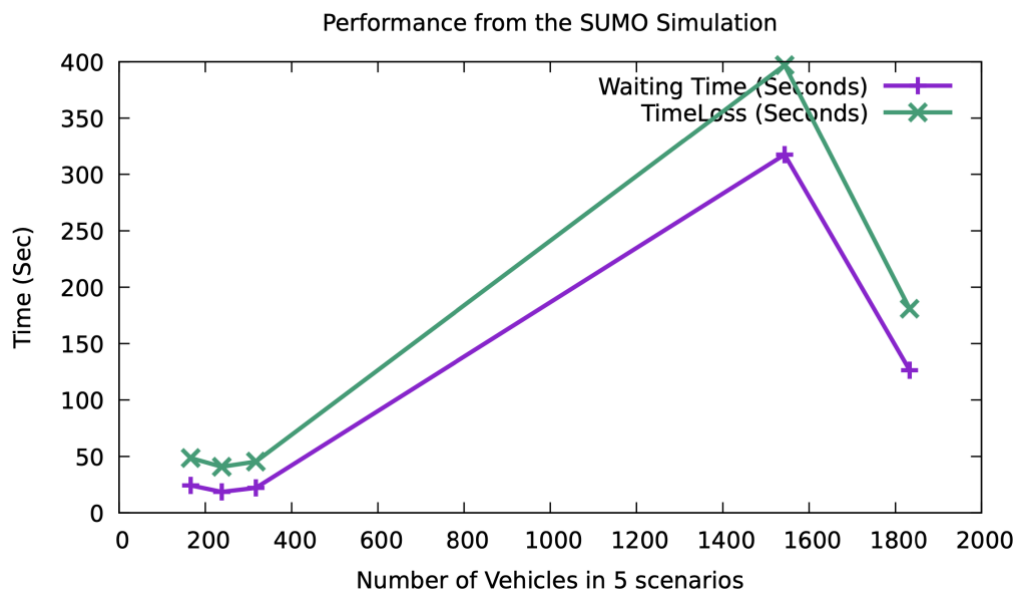


Figure 18 : Time loss & Waiting time

Figure 19 shows the departure delay and speed in kmph. The average speed of the vehicles is calculated and its around 9kmph as the input speed was too less in the roads. Also figure 3 shows the average delay in departure of the vehicles. The vehicles usually start after a delayed departure. This is also important while analysis characteristics of the vehicular networks.

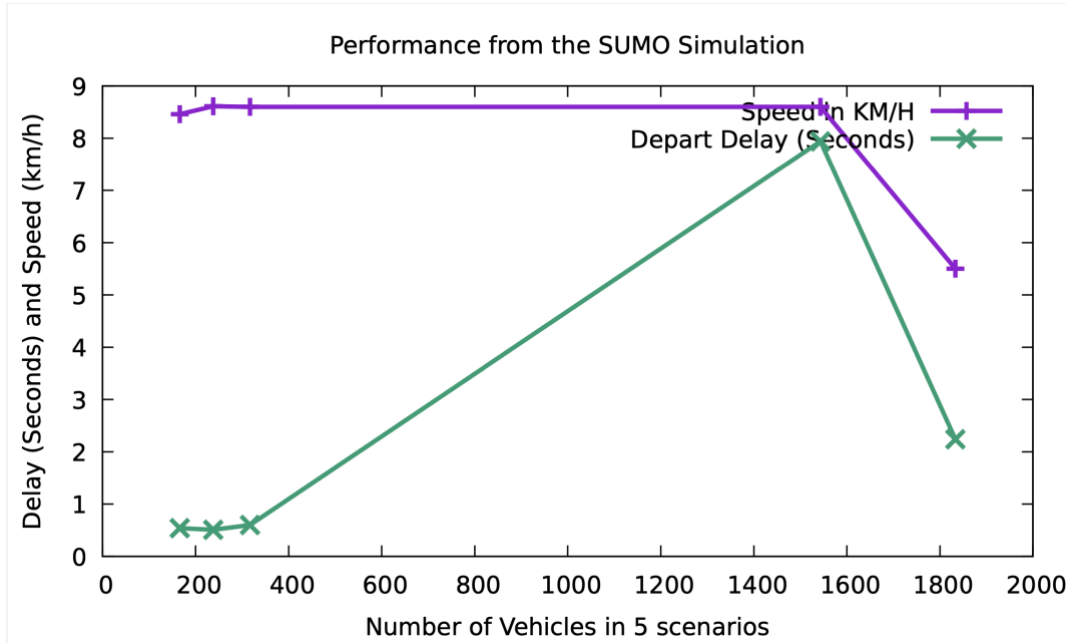


Figure 19: Speed & Depart Delay

Figure 20 is based on the travelling time of the vehicles which was recorded using the network simulation software. The travelling time is higher in the SUMO than the networks. This is due to the following conditions like mobility of the network and also the calculation is based on the approximation of Duration, End to End Delay between the vehicles. But using VANET we can see that the travel time is less because using VANET let cars communicates between each other, so driver can avoid any conditions in the road like jams or accidents and others.

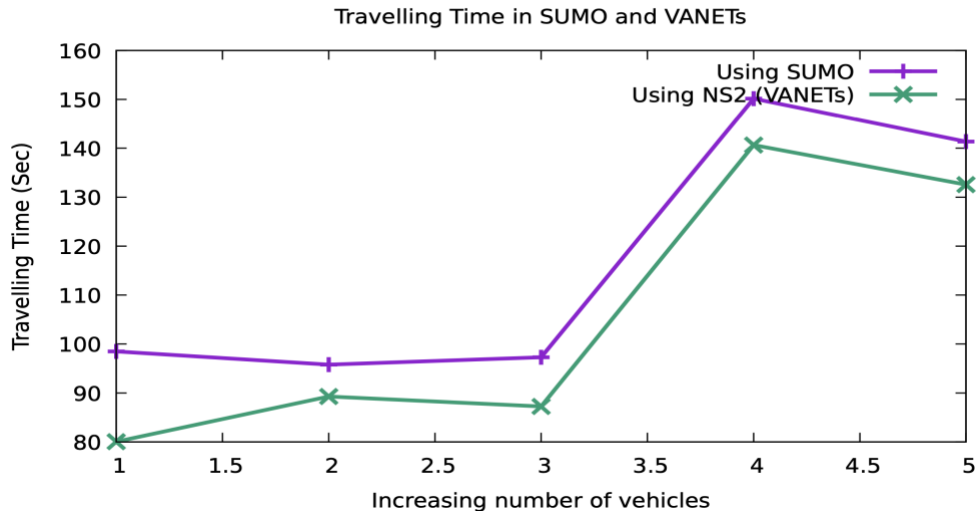


Figure 20 :Travelling Time with SUMO and VANETs

Since in the simulation, there were teleports during the Scenario 4 which indicates that the vehicles in the road are travelling more and hence the average travelling time is high. This is due to the collision encountered in the scenario. This collision data is shown in table 3 and figure 21 below :

Table 3: collision data

Scenario	N.Vehicles	Jams	Wrong Lane	Yield
4	1543	32	1	19
5	1833	2	0	7

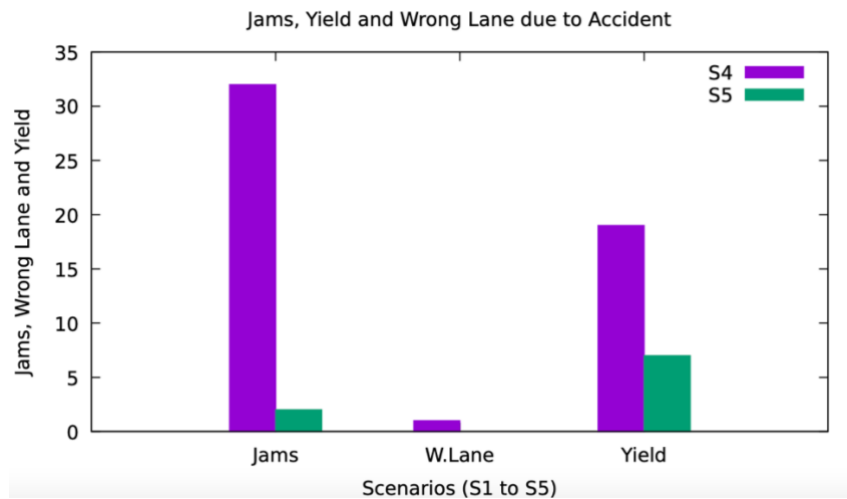


Figure 21: Collision graph

We got the results of teleporting while collision on the roads leads to Traffic Jams, taking Wrong Lane and Yield. As per the table given below, we have arrived at a simulation from sumo these two scenarios. These collisions usually occurs when the number of vehicles is in huge numbers. In our scenario, the vehicle used in S4 and S5 are namely 1500+ and 1833 respectively. Hence there are more jams, yields and wrong lanes. The traffic jams was in 32 numbers and some vehicles take the wrong lane other than what is intended for and there was a yielding value of 19.

Emission classes and standards also affect the smart driving assistance system, as the driver has to ride the vehicle with optimal speed, latest or recent emission norms, routes for optimization and other road conditions to optimize the fuel consumption of the vehicles. In our work, we have taken three types of vehicles as indicated earlier namely Passenger Cars, Motor Bikes and Trucks. All the three emits the following gases based on the condition of the road, emission norms and vehicle conduction. But a smart driving system will optimize all these. We have done the simulations in SUMO to handle these emission classes that ensures an error free smart driving assistance system.

As per the scenarios, we have generated totally 5 scenarios with varying number of vehicles namely motorbikes, cars and trucks. The average emission of gases in each of the scenarios is shown in the following figures.

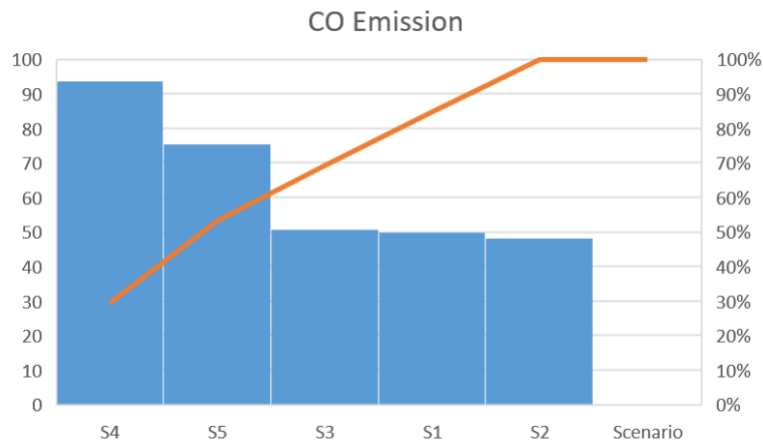


Figure 22: CO Emission

In the Carbon Monoxide emission, its high in scenario 4, as the vehicles where waiting for a longer period of time in the traffic and in the accident, some of the vehicles have turned the wrong lane or jams or yield. So the CO emission is too high in the S4 compared to other scenarios. S2 is optimal that its CO emission is almost as low as 49%. Hydro Carbons also is more in S4, as HC are emitted only by the LDVs but still its high in the S4 segment, since the vehicles are waiting in the traffic due to the accidents.

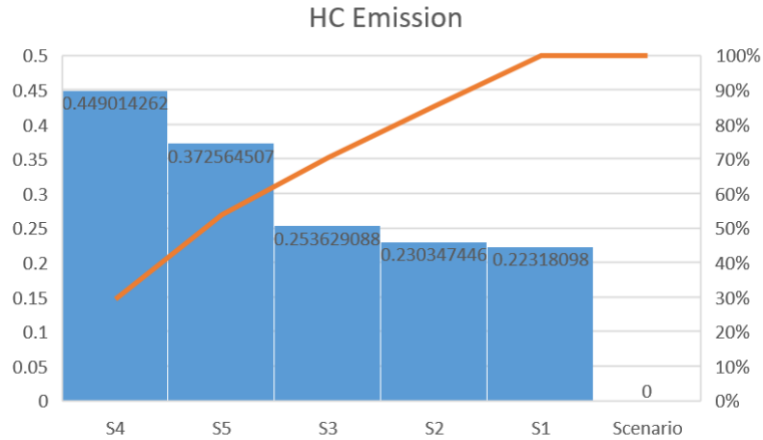


Figure 23: HC Emission

Carbon dioxide, Fuel Consumption, NOx and PMx are high in scenario 1 as the vehicles are less in numbers and they are freely moving in the lanes that tells its high in this scenario. But those emission are too controlled in the S4 and S5 though there are more vehicles in these scenarios that indicates that the vehicles are yielding or jamming. Hence the CO and HC are high, and others are very less. So, the emission and the fuel consumption decides the smart driving assistance system. All these values are generated from SUMO, and they are random in nature. However, in reality, the drivers can be smart if they drive the vehicle in such a way that it can optimize the fuel economy and emits lesser emissions if the vehicles follow the emission norms.

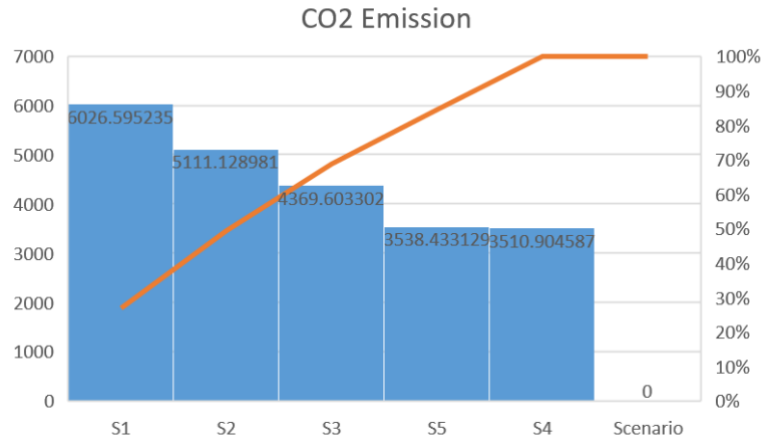


Figure 24: CO2 Emission

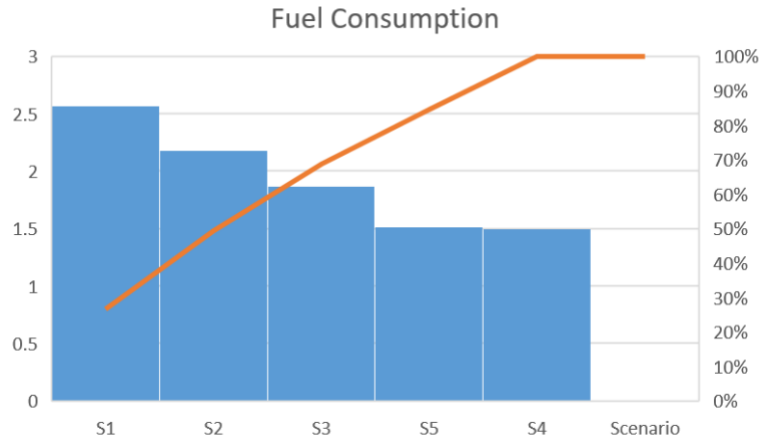


Figure 25: Fuel Consumption

Figure 26 shows the communication between two cars that indicates there is a collision in the network. Also this indicates the driver is notified with a message through the network that there is a collision in the network.

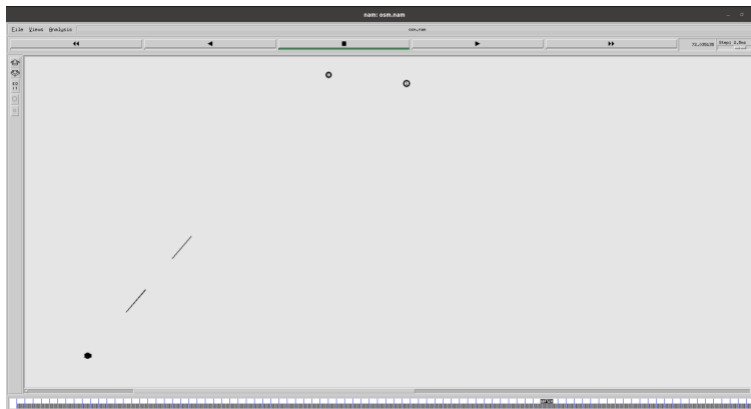


Figure 26: Communication between cars in VANET

Finally, we collect all these types of data to use them for smart awareness system for drivers through mobile phone or through head up display in the car so the driver can receive real time information to take the right decision in the right time and right place and as well to avoid any risks during the driving.

Chapter 6

6.1 CONCLUSION & FUTURE WORK

The importance of traffic information is one of the most important aspects of the definition and creation of ITS solutions. Travelers need accurate and up-to-date traffic information in order to arrive at their destinations more quickly and safely. In this region, the SASD stands out among the ITS. SASD gathers data from its surroundings and produces awareness from it, which is then broadcast to travelers based on their individual needs, enhancing their level of cognition. In this way, the traveler can use it in his decision-making process when it comes to optimum roads, weather conditions, and minimized environmental impacts, among other things.

Simulation has been shown to be a useful tool for analyzing and developing novel traffic solutions in socio-technical structures. Over the past few decades, traffic systems have undergone significant improvements, and travelers have seen a transformation in the way trips are organized in urban networks. As a result, in light of the current traffic situation in most developed countries, it is now critical to promote new transportation strategies based on cutting edge technology for Future Urban Transportation (FUT). In the latest vision of urban systems, the user is now the central figure. Traditional traffic simulation packages fail to model and reflect all aspects of human behavior in this new setting. Most simulators use a macro/microscopic approach to improve the portrayal of traffic flow and management rather than traveler behavior in general.

V2V (vehicle-to-vehicle) and V2I (vehicle-to-infrastructure) communications are critical components of the ITS architecture. Vehicular Ad-Hoc Networks (VANET) are a form of vehicle-to-vehicle communication that has a number of benefits, including reducing car accidents, reducing traffic congestion, lowering fuel consumption and pollution, and so on. The IEEE 802.11p specification has standardized VANET architecture. Ad-hoc network simulators can simulate ad-hoc networks, but they can't simulate city traffic. To address this problem, this thesis looked into the Veins system, which is used to run a traffic (SUMO) and network (OMNET++) simulator in parallel, simulating practical traffic in Hasselt, Belgium, as an ad-hoc network.

Future work

Taking into account the resulting architecture, a wide range of applications is possible. New and more accurate visualization techniques may be combined to aim for practical simulations. In addition, a potential integration with a Serious Game driver simulator will be used to provide a Human-in-the-Loop simulation, which will be used to evaluate and verify the user's decision-making process when obtaining information. Connecting our system to a database that contains

demographic data about the population in order to better reflect the SASD is also a potential project. Furthermore, a database of additional networks, edges, or nodes may provide an agent delegate of the simulation information, allowing the network topology to be expanded or modified on the fly, and the simulation to be resumed from where it was previously stopped. As a result, we've added another dimension to our simulation. Researchers can study and build better and more complicated algorithms for traffic management or route planning using the MAS development framework.

Reference:

World Health Organization. (n.d.). *Road traffic injuries*. World Health Organization. <https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries>.

Eurostat. (2020, July 1). *Over 23 000 victims of road accidents in the EU in 2018*. Over 23 000 victims of road accidents in the EU in 2018 - Products Eurostat News - Eurostat. <https://ec.europa.eu/eurostat/web/products-eurostat-news/-/DDN-20200701-1>.

4 Ways Road Crashes Impact the Economy. Together For Safer Roads. (2016, September 20). <https://www.togetherforsaferroads.org/4-ways-road-crashes-impact-the-economy/>.

Sarah O'Meara Jul. 17, 2020. (2020, July 17). *The road to reducing traffic accidents in China*. Science. <https://www.sciencemag.org/features/2020/07/road-reducing-traffic-accidents-china>.

KHAN, I. (2020, August 25). *Vision Zero*. Mobility and transport - European Commission. https://ec.europa.eu/transport/road_safety/what-we-do_en.

Hololei, 2021. [online] Ec.europa.eu. Available at: <https://ec.europa.eu/transport/sites/transport/files/2018-transport-in-the-eu-current-trends-and-issues.pdf> [Accessed 21 April 2018].

Rodrigue, D. J.-P., & Notteboom, D. T. (n.d.). *Contents*. The Geography of Transport Systems. <https://transportgeography.org/contents/chapter3/transportation-and-economic-development/>.

Woodrow, & Thomas. (1998). Google. <https://books.google.com.om/books?hl=ar>.

Li, J., Tang, S., Wang, X., & Wang, F.-Y. (2007). A Software Architecture for Artificial Transportation Systems ; Principles and Framework. *2007 IEEE Intelligent Transportation Systems Conference*. <https://doi.org/10.1109/itsc.2007.4357814>

Jamal, H. (2017, March 1). *Haseeb Jamal*. Objectives, Advantages and Applications. <https://www.aboutcivil.org/advanced-traveller-information-system-ATIS.html>.

Kachan, D. (2010). *Integration of NS-3 with MATLAB/Simulink*. <http://tu.diva-portal.org/smash/get/diva2:1024227/FULLTEXT01.pdf>.

Kaths, J., Krause, S., & Busch, F. (2017). *Integrated simulation of microscopic traffic flow and vehicle dynamics*. mediatum. <https://mediatum.ub.tum.de/doc/1327520/file.pdf>.

Chenxi et al (2012). Evaluation of CACC string stability using SUMO, Simulink, and OMNeT++. EURASIP Journal on Wireless Communications and Networking

Ibrahim, A., Belagal Math, C., Goswami, D., & Basten, T. (2018). *Co-simulation Framework for Control, Communication and Traffic for Vehicle Platoons*. IEEE Xplore. <https://ieeexplore.ieee.org/document/8491839>.

Samaoui, S., & Mansouri, W. (2015). *Network Simulator*. Network Simulator - an overview | ScienceDirect Topics. <https://www.sciencedirect.com/topics/computer-science/network-simulator>.

Kabir, M., Islam, S. and Hossain, J., 2014. *Detail Comparison of Network Simulators*. [online] Ijser.org. Available at: <<https://www.ijser.org/researchpaper/Detail-Comparison-of-Network-Simulators.pdf>>

Pan, J. (n.d.). *A Survey of Network Simulation Tools: Current Status and Future Developments*. A Survey of Network Simulation Tools: Current Status and Future Development. <https://www.cse.wustl.edu/~jain/cse567-08/ftp/simtools/index.html>.

Jawandhiya, P., Thakare, P. and M. Ghonge, M., 2013. *Open-Source Network Simulation Tools An Overview*. [online] researchgate.net. Available at: <https://www.researchgate.net/publication/249998863_Open-Source_Network_Simulation_Tools_An_Overview>

Fiore, M., Harri, J., Filali, F., & Bonnet, C. (2007). Vehicular Mobility Simulation for VANETs. *40th Annual Simulation Symposium (ANSS07)*. doi:10.1109/anss.2007.44

Hartenstein, H., & Laberteaux, K. (2010). *VANET: Vehicular applications and inter-networking technologies*. Chichester: Wiley.

Sommer, C., German, R., & Dressler, F. (2011). Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis. *IEEE Transactions on Mobile Computing*,10(1), 3-15. doi:10.1109/tmc.2010.133

Krajzewicz, D. (2010). Traffic Simulation with SUMO – Simulation of Urban Mobility. *Fundamentals of Traffic Simulation International Series in Operations Research & Management Science*,269-293. doi:10.1007/978-1-4419-6142-6_7

Wilkie, D. (2015). *Virtualized Traffic at Metropolitan Scales*. Frontiers. <https://www.frontiersin.org/articles/10.3389/frobt.2015.00011/full#h10>

Hallerbach, S., Xia, Y., Eberle, U., and Koester, F., "Simulation-Based Identification of Critical Scenarios for Cooperative and Automated Vehicles," *SAE Intl. J CAV* 1(2):93-106, 2018, <https://doi.org/10.4271/2018-01-1066>.

Scharfe, M., Zeeb, K., & Russwinkel, N. (2020). The Impact of Situational Complexity and Familiarity on Takeover Quality in Uncritical Highly Automated Driving Scenarios. *Information*, 11(2), 115. doi: 10.3390/info11020115

S. Al-Sultan, A. H. Al-Bayatti and H. Zedan, "Context-Aware Driver Behavior Detection System in Intelligent Transportation Systems," in *IEEE Transactions on Vehicular Technology*, vol. 62, no. 9, pp. 4264-4275, Nov. 2013, doi: 10.1109/TVT.2013.2263400.

Agarwal, Y., Jain, K., & Karabasoglu, O. (2018). Smart vehicle monitoring and assistance using cloud computing in vehicular Ad Hoc networks. Retrieved 8 June 2021, from

Samaoui, S. (2015). Network Simulator - an overview | ScienceDirect Topics. Retrieved 8 June 2021, from <https://www.sciencedirect.com/topics/computer-science/network-simulator>

Clarification

This master thesis was written during the COVID-19 crisis in 2021. This global health crisis has had an impact on the (writing) process, the research activities and the research results that are at the basis of this thesis because:

- I attempted several times to get a visa and the permission to visit Hasselt University to carry out my research but without success.*
- Several important resources are available on libraries, which were closed during the pandemic*
- There was no access to the labs.*
- The necessary data collection was difficult.*
- Reaching and connecting to other students with same path were very difficult.*
- Communications with my supervisors were always online. This was not always convenient to explain my progress.*

APPENDIX

L1 L2 L3 L4 - Left Junctions or Left Nodes

M1 M2 M3 M4 - Middle Junctions or Middle Nodes

R1 R2 R3 R4 - Right Junctions or Right nodes.

These things are written to a file called myown.nod.xml

Edges or Roads

rl1m1 means right side connection between nodes L1 and M1.

ll1m1 means left side connection between nodes L1 and M1

dl1l2 means downwards connection between L1 and L2

ul1l2 means upwards connection between L1 and L2

d- downward

u - upward

l -left

r- right

The roads or the edges information are written to myown.edg.xml file.

\

Edges have priorities from 1, 2, 3, the higher the priority, higher is the number and more vehicles will move towards higher priority roads.

Two files are created namely myown.nod.xml and myown.edg.xml

Convert these in to a myown.net.xml file.

```
$ netconvert --node-files=myown.nod.xml --edge-files=myown.edg.xml --output-  
file=myown.net.xml
```

Once you get a success message, the conversion is successful and it creates a file called myown.net.xml

```
$ polyconvert --net-file myown.net.xml -o myown.poly.xml
```

To create a route file. we need to use randomTrips.py file.

```
$ python /home/meamrashid/sumo/tools/randomTrips.py -n myown.net.xml -r myown.rou.xml -e  
100 -l
```

Success message indicates its successful and it generated three files

namely a trips.trip.xml file

rou.xml file

rou.alt.xml file.

```
<configuration>  
  <input>  
    <net-file value="myown.net.xml"/>  
  <route-files value="myown.rou.xml"/>  
    <additional-files value="myown.poly.xml"/>  
  </input>  
<time>  
  <begin value="0"/>  
  <end value="100"/>  
  <step-length value="0.1"/>  
</time>
```

</configuration>

```
$ sumo -c myown.sumocfg --summary myown_summary.xml
```

To run the simulation in a GUI

```
$ sumo-gui -c myown.sumocfg
```

Most of the files are xml based, but we can convert those files to csv files for plotting graphs and other results.

There is an option in sumo itself called
xml2csv.py (that converts xml file to csv)
CSV - Comma Separated Values.

Use the following commands.

```
$ cd /home/meamrashid/sumo/tools/xml  
$ python xml2csv.py /home/meamrashid/myownsumomap/myown_statistics.xml
```

To get emission, collision and statistics file in the sumo command, use the following command.

```
$ sumo -c myown.sumocfg --summary summary.xml --device.battery.track-fuel true --collision-  
output myown_collision.xml --statistic-output myown_statistics.xml --emission-output  
myown_emission.xml
```

Floating Car Data (fcd)

```
$ sumo -c myown.sumocfg --fcd-output myown.sumo.xml
```

VANET

If you want to include VANET (Connected Cars), then we have to use TraceExporter to export our xml files to ns3 or ns2 or OMNETpp

All the three are network simulators

ns2 - TCL and C++

ns3 - Python and C++
OMNETPP - Java and C++

This command is used to create our xml files to TCL(Tool Command Language) Files that can be run under ns2 or ns3.

```
$ python /home/meamrashid/sumo/tools/traceExporter.py --fcd-input myown.sumo.xml --  
ns2config-output myown.tcl --ns2activity-output myown_activity.tcl --ns2mobility-output  
myown_mobility.tcl
```

Now move to OSM (Open Street Map)
Open a terminal

```
$ cd sumo/tools/  
$ python osmWebWizard.py  
It will open a browser with a OSM (Map)
```

.....
Summary:

Node type statistics:

Unregulated junctions : 0
Dead-end junctions : 120
Priority junctions : 290
Right-before-left junctions : 77
Traffic light junctions : 9

Network boundaries:

Original boundary : 5.32,50.92,5.35,50.93
Applied offset : -662717.62,-5643764.21
Converted boundary : 0.00,0.00,2374.72,1394.76

OSM Performance

Simulation version 1.9.2 started with time: 0.00

Warning: Vehicle 'veh210' performs emergency braking with decel=-9.00 wished=4.50 severity=1.00, time=3071.00.

Simulation ended at time: 3772.00

Reason: All vehicles have left the simulation.

Performance:

Duration: 1.67s

Real time factor: 2264.11

UPS: 14656.062425

Vehicles:

Inserted: 166

Running: 0

Waiting: 0

Statistics (avg):

RouteLength: 1180.50

Speed: 8.46

Duration: 147.09

WaitingTime: 24.23

TimeLoss: 48.61

DepartDelay: 0.54

DijkstraRouter answered 166 queries and explored 125.90 edges on average.

DijkstraRouter spent 0.02s answering queries (0.13ms on average).

To plot the characteristics: We need data for 5 scenarios with varying number of vehicles

Scenario3

Summary:

Node type statistics:

Unregulated junctions : 0

Dead-end junctions : 120

Priority junctions : 290

Right-before-left junctions : 77

Traffic light junctions : 9

Network boundaries:

Original boundary : 5.32,50.92,5.35,50.93

Applied offset : -662717.62,-5643764.21

Converted boundary : 0.00,0.00,2374.72,1394.76

Scenario4

Summary:

Node type statistics:

Unregulated junctions : 0
Dead-end junctions : 166
Priority junctions : 522
Right-before-left junctions : 142
Traffic light junctions : 11

Network boundaries:

Original boundary : 5.31,50.92,5.36,50.94
Applied offset : -662662.16,-5643764.21
Converted boundary : 0.00,0.00,3126.85,1801.84

Scenario 5

Summary:

Node type statistics:

Unregulated junctions : 0
Dead-end junctions : 166
Priority junctions : 522
Right-before-left junctions : 142
Traffic light junctions : 11

Network boundaries:

Original boundary : 5.31,50.92,5.36,50.94
Applied offset : -662662.16,-5643764.21
Converted boundary : 0.00,0.00,3126.85,1801.84

Scenario 1 - SUMO Simulation

OSM Performance

Simulation version 1.9.2 started with time: 0.00

Warning: Vehicle 'veh210' performs emergency braking with decel=-9.00 wished=4.50 severity=1.00, time=3071.00.

Simulation ended at time: 3772.00

Reason: All vehicles have left the simulation.

Performance:

Duration: 1.67s
Real time factor: 2264.11
UPS: 14656.062425

Vehicles:

Inserted: 166

Running: 0
Waiting: 0
Statistics (avg):
RouteLength: 1180.50
Speed: 8.46
Duration: 147.09
WaitingTime: 24.23
TimeLoss: 48.61
DepartDelay: 0.54

DijkstraRouter answered 166 queries and explored 125.90 edges on average.
DijkstraRouter spent 0.02s answering queries (0.13ms on average).

S2

Performance:
Duration: 1.87s
Real time factor: 1998.39
UPS: 17429.260450
Vehicles:
Inserted: 238
Running: 0
Waiting: 0
Statistics (avg):
RouteLength: 1143.35
Speed: 8.71
Duration: 136.65
WaitingTime: 18.58
TimeLoss: 40.86
DepartDelay: 0.51

DijkstraRouter answered 238 queries and explored 120.55 edges on average.
DijkstraRouter spent 0.02s answering queries (0.09ms on average).

S3

Performance:
Duration: 2.33s
Real time factor: 1623.87
UPS: 19419.922714
Vehicles:

Inserted: 317
Running: 0
Waiting: 0
Statistics (avg):
RouteLength: 1162.43
Speed: 8.60
Duration: 142.68
WaitingTime: 22.08
TimeLoss: 45.40
DepartDelay: 0.60

DijkstraRouter answered 317 queries and explored 122.39 edges on average.
DijkstraRouter spent 0.04s answering queries (0.11ms on average).

S4

Simulation version 1.9.2 started with time: 0.00

Warning: Teleporting vehicle 'moto15'; waited too long (yield), lane='511821333_0', time=862.00.

Warning: Vehicle 'moto15' ends teleporting on edge '511821332', time 862.00.

Warning: Teleporting vehicle 'veh202'; waited too long (jam), lane='245515620#1_0', time=887.00.

Warning: Vehicle 'veh202' ends teleporting on edge '245515620#3', time 887.00.

Warning: Teleporting vehicle 'veh136'; waited too long (yield), lane='511821333_0', time=1166.00.

Warning: Vehicle 'veh136' ends teleporting on edge '511821332', time 1166.00.

Warning: Teleporting vehicle 'veh192'; waited too long (jam), lane='245515620#1_0', time=1191.00.

Warning: Vehicle 'veh192' ends teleporting on edge '245515620#3', time 1191.00.

Warning: Teleporting vehicle 'veh280'; waited too long (jam), lane='511821331#0_0', time=1236.00.

Warning: Teleporting vehicle 'veh229'; waited too long (yield), lane='513851435_0', time=1236.00.

Warning: Teleporting vehicle 'truck51'; waited too long (yield), lane='511821331#1_0', time=1236.00.

Warning: Vehicle 'veh280' ends teleporting on edge '511821331#1', time 1236.00.

Warning: Vehicle 'veh229' ends teleporting on edge '511821330#2', time 1239.00.

Warning: Vehicle 'truck51' ends teleporting on edge '53004396#0', time 1243.00.

Warning: Teleporting vehicle 'truck110'; waited too long (jam), lane='245515620#1_0', time=1973.00.

Warning: Vehicle 'truck110' ends teleporting on edge '245515620#3', time 1973.00.

Warning: Teleporting vehicle 'truck166'; waited too long (jam), lane=':1813633321_6_0', time=2039.00.
Warning: Vehicle 'truck166' ends teleporting on edge '116795272', time 2041.00.
Warning: Teleporting vehicle 'veh612'; waited too long (yield), lane='511821332_0', time=2107.00.
Warning: Teleporting vehicle 'veh463'; waited too long (jam), lane=':223673500_1_0', time=2110.00.
Warning: Vehicle 'veh463' ends teleporting on edge '511821332', time 2113.00.
Warning: Vehicle 'veh612' teleports beyond arrival edge '513851435', time 2113.00.
Warning: Teleporting vehicle 'veh625'; waited too long (yield), lane='511821333_0', time=2115.00.
Warning: Vehicle 'veh625' ends teleporting on edge '-316852438', time 2120.00.
Warning: Teleporting vehicle 'moto85'; waited too long (jam), lane='245515620#1_0', time=2421.00.
Warning: Teleporting vehicle 'truck110'; waited too long (yield), lane='511821333_0', time=2421.00.
Warning: Vehicle 'truck110' ends teleporting on edge '511821332', time 2421.00.
Warning: Vehicle 'moto85' ends teleporting on edge '245515620#3', time 2422.00.
Warning: Teleporting vehicle 'moto82'; waited too long (yield), lane='513851435_0', time=2476.00.
Warning: Vehicle 'moto82' ends teleporting on edge '182038028', time 2476.00.
Warning: Teleporting vehicle 'veh744'; waited too long (jam), lane='511821331#0_0', time=2479.00.
Warning: Vehicle 'veh744' ends teleporting on edge '511821331#1', time 2479.00.
Warning: Vehicle 'veh744' performs emergency braking with decel=-9.00 wished=4.50 severity=1.00, time=2480.00.
Warning: Teleporting vehicle 'veh1060'; waited too long (jam), lane='20822017#4_0', time=2523.00.
Warning: Vehicle 'veh1060' ends teleporting on edge '53004396#0', time 2532.00.
Warning: Teleporting vehicle 'veh519'; waited too long (jam), lane=':223673500_1_0', time=2538.00.
Warning: Vehicle 'veh519' ends teleporting on edge '511821330#0', time 2538.00.
Warning: Teleporting vehicle 'veh1028'; waited too long (jam), lane='623915972#1_1', time=2542.00.
Warning: Vehicle 'veh1028' ends teleporting on edge '208584391#0', time 2542.00.
Warning: Teleporting vehicle 'veh866'; waited too long (jam), lane='208584392_1', time=2554.00.
Warning: Teleporting vehicle 'veh1005'; waited too long (jam), lane='570096339#0_1', time=2585.00.
Warning: Vehicle 'veh1005' ends teleporting on edge '116795269#0', time 2594.00.

Warning: Teleporting vehicle 'truck215'; waited too long (wrong lane), lane='511821329_0', time=2619.00.

Warning: Teleporting vehicle 'truck214'; waited too long (jam), lane='570096339#1_1', time=2620.00.

Warning: Vehicle 'veh866' ends teleporting on edge '208584391#0', time 2648.00.

Warning: Vehicle 'truck214' ends teleporting on edge '623915972#1', time 2658.00.

Warning: Teleporting vehicle 'veh1128'; waited too long (jam), lane=':490607642_0_1', time=2711.00.

Warning: Vehicle 'veh1128' ends teleporting on edge '623915972#0', time 2713.00.

Warning: Teleporting vehicle 'truck197'; waited too long (jam), lane=':2273191310_9_0', time=2714.00.

Warning: Teleporting vehicle 'truck225'; waited too long (jam), lane='623915972#0_0', time=2718.00.

Warning: Teleporting vehicle 'veh1141'; waited too long (yield), lane='18008107#1_0', time=2718.00.

Warning: Teleporting vehicle 'veh1127'; waited too long (jam), lane=':5891482336_4_0', time=2719.00.

Warning: Vehicle 'truck197' ends teleporting on edge '208584392', time 2719.00.

Warning: Vehicle 'veh1127' ends teleporting on edge '671699517', time 2719.00.

Warning: Vehicle 'truck225' ends teleporting on edge '208584391#0', time 2722.00.

Warning: Vehicle 'veh1141' ends teleporting on edge '116795269#0', time 2723.00.

Warning: Vehicle 'truck215' ends teleporting on edge '53004431', time 2743.00.

Warning: Teleporting vehicle 'veh1033'; waited too long (jam), lane='570096339#0_1', time=2891.00.

Warning: Teleporting vehicle 'truck222'; waited too long (jam), lane='570096339#1_1', time=2921.00.

Warning: Vehicle 'veh1033' ends teleporting on edge '623915972#1', time 2956.00.

Warning: Vehicle 'truck222' ends teleporting on edge '623915972#1', time 2958.00.

Warning: Teleporting vehicle 'veh1216'; waited too long (yield), lane='570096338_1', time=3030.00.

Warning: Teleporting vehicle 'veh1223'; waited too long (jam), lane=':490607642_0_1', time=3063.00.

Warning: Vehicle 'veh1223' ends teleporting on edge '623915972#1', time 3066.00.

Warning: Teleporting vehicle 'truck194'; waited too long (jam), lane=':5891482336_4_0', time=3069.00.

Warning: Teleporting vehicle 'truck228'; waited too long (yield), lane='18008107#1_0', time=3069.00.

Warning: Vehicle 'truck194' ends teleporting on edge '671699517', time 3069.00.

Warning: Teleporting vehicle 'veh1164'; waited too long (jam), lane='18008107#0_0', time=3072.00.

Warning: Teleporting vehicle 'moto71'; waited too long (jam), lane='360767722#0_0', time=3077.00.

Warning: Teleporting vehicle 'veh1241'; waited too long (jam), lane=':490607642_0_0', time=3084.00.

Warning: Vehicle 'veh1164' ends teleporting on edge '208584392', time 3084.00.

Warning: Vehicle 'truck228' ends teleporting on edge '623915972#1', time 3086.00.

Warning: Vehicle 'veh1241' ends teleporting on edge '623915972#1', time 3087.00.

Warning: Teleporting vehicle 'veh1235'; waited too long (jam), lane='-127123963#1_0', time=3093.00.

Warning: Teleporting vehicle 'truck255'; waited too long (jam), lane='-221071908_0', time=3098.00.

Warning: Vehicle 'veh1216' ends teleporting on edge '360767722#0', time 3099.00.

Warning: Teleporting vehicle 'veh1127'; waited too long (jam), lane='116795269#1_1', time=3101.00.

Warning: Vehicle 'veh1127' ends teleporting on edge '208584392', time 3101.00.

Warning: Teleporting vehicle 'veh1250'; waited too long (jam), lane=':2273191310_6_0', time=3102.00.

Warning: Vehicle 'truck255' ends teleporting on edge '-62733572', time 3102.00.

Warning: Vehicle 'veh1250' ends teleporting on edge '116795269#0', time 3104.00.

Warning: Teleporting vehicle 'veh1240'; waited too long (jam), lane='623915972#0_0', time=3158.00.

Warning: Vehicle 'veh1240' ends teleporting on edge '623915972#1', time 3158.00.

Warning: Teleporting vehicle 'veh1345'; waited too long (yield), lane=':1064221065_5_0', time=3167.00.

Warning: Vehicle 'veh1345' ends teleporting on edge '53004396#0', time 3170.00.

Warning: Vehicle 'moto71' ends teleporting on edge '671699518', time 3175.00.

Warning: Vehicle 'veh1235' ends teleporting on edge '116795269#1', time 3207.00.

Warning: Teleporting vehicle 'veh1394'; waited too long (yield), lane='-23582528#0_0', time=3242.00.

Warning: Vehicle 'veh1394' ends teleporting on edge '-221071908', time 3252.00.

Warning: Teleporting vehicle 'truck259'; waited too long (yield), lane='18008107#1_0', time=3370.00.

Warning: Vehicle 'truck259' ends teleporting on edge '116795269#0', time 3375.00.

Warning: Teleporting vehicle 'veh1302'; waited too long (yield), lane='511821333_0', time=3473.00.

Warning: Teleporting vehicle 'veh1382'; waited too long (jam), lane='245515620#1_0', time=3475.00.

Warning: Vehicle 'veh1382' ends teleporting on edge '245515620#3', time 3476.00.

Warning: Vehicle 'veh1302' ends teleporting on edge '-316852438', time 3478.00.

Warning: Teleporting vehicle 'veh1578'; waited too long (yield), lane='234182324#1_0', time=3647.00.

Warning: Teleporting vehicle 'veh1329'; waited too long (yield), lane='18008107#1_0', time=3671.00.

Warning: Teleporting vehicle 'veh1211'; waited too long (jam), lane='18008107#0_0', time=3674.00.

Warning: Vehicle 'veh1329' ends teleporting on edge '671699518', time 3674.00.

Warning: Vehicle 'veh1578' ends teleporting on edge '242746088#0', time 3674.00.

Warning: Vehicle 'veh1211' ends teleporting on edge '116795269#0', time 3680.00.

Warning: Teleporting vehicle 'veh1560'; waited too long (yield), lane='18008107#1_0', time=3972.00.

Warning: Vehicle 'veh1560' ends teleporting on edge '116795269#0', time 3977.00.

Warning: Teleporting vehicle 'veh1562'; waited too long (yield), lane='570096339#1_0', time=4033.00.

Warning: Vehicle 'veh1562' ends teleporting on edge '127123963#0', time 4035.00.

Simulation ended at time: 5126.00

Reason: All vehicles have left the simulation.

Performance:

Duration: 32.37s

Real time factor: 158.361

UPS: 26080.107510

Vehicles:

Inserted: 1543

Running: 0

Waiting: 0

Teleports: 52 (Jam: 32, Yield: 19, Wrong Lane: 1)

Statistics (avg):

RouteLength: 1817.16

Speed: 5.50

Duration: 547.11

WaitingTime: 317.45

TimeLoss: 396.96

DepartDelay: 7.94

DijkstraRouter answered 1649 queries and explored 285.05 edges on average.

DijkstraRouter spent 0.51s answering queries (0.31ms on average).

S5

Warning: Teleporting vehicle 'truck123'; waited too long (yield), lane='234182324#1_0', time=1666.00.
Warning: Vehicle 'truck123' ends teleporting on edge '242746088#0', time 1694.00.
Warning: Teleporting vehicle 'veh1146'; waited too long (yield), lane='234182324#1_0', time=2529.00.
Warning: Vehicle 'veh1146' ends teleporting on edge '242746088#0', time 2556.00.
Warning: Teleporting vehicle 'veh1894'; waited too long (yield), lane='-476160628#2_0', time=3700.00.
Warning: Vehicle 'veh1894' teleports beyond arrival edge '476160628#1', time 3701.00.
Warning: Teleporting vehicle 'veh1912'; waited too long (yield), lane='476160628#1_0', time=3707.00.
Warning: Vehicle 'veh1912' ends teleporting on edge '106373416#0', time 3707.00.
Warning: Teleporting vehicle 'veh1955'; waited too long (yield), lane='511821333_0', time=3863.00.
Warning: Vehicle 'veh1955' ends teleporting on edge '511821332', time 3863.00.
Warning: Teleporting vehicle 'veh2083'; waited too long (jam), lane='245515620#1_0', time=3877.00.
Warning: Vehicle 'veh2083' ends teleporting on edge '245515620#3', time 3877.00.
Warning: Teleporting vehicle 'veh2005'; waited too long (yield), lane='513851435_0', time=3907.00.
Warning: Vehicle 'veh2005' ends teleporting on edge '182038028', time 3907.00.
Warning: Teleporting vehicle 'veh2021'; waited too long (jam), lane='511821331#0_0', time=3908.00.
Warning: Vehicle 'veh2021' ends teleporting on edge '511821331#1', time 3909.00.
Warning: Teleporting vehicle 'veh1927'; waited too long (yield), lane='511821332_0', time=4648.00.
Warning: Vehicle 'veh1927' ends teleporting on edge '513851435', time 4652.00.
Simulation ended at time: 4797.00
Reason: All vehicles have left the simulation.
Performance:
Duration: 23.03s
Real time factor: 208.275
UPS: 25656.825287
Vehicles:
Inserted: 1833
Running: 0
Waiting: 0
Teleports: 9 (Jam: 2, Yield: 7)
Statistics (avg):
RouteLength: 1723.05

Speed: 6.90
Duration: 322.38
WaitingTime: 126.36
TimeLoss: 180.91
DepartDelay: 2.24

DijkstraRouter answered 1850 queries and explored 281.65 edges on average.
DijkstraRouter spent 0.58s answering queries (0.31ms on average).

To plot the graph, i use a software called gnuplot
For this you need a plot file namely *.plt and data files.

From the 5 scenario we have tabulated Number of Vehicles, AVerage Speed, Waiting Time, Time Loss and Depart Delay.

The name of the following file is sumo_scenario.dat

```
166 8.46 24.23 48.6 0.54
238 8.61 18.58 40.86 0.51
317 8.60 22.08 45.4 0.6
1543 8.6 317.45 396.96 7.94
1833 5.5 126.36 180.91 2.24
```

and we have a plot file called plot.plt

```
set terminal pdf
```

```
set output "scenario_sumo.pdf"
```

```
set title "Performance from the SUMO Simulation"
```

```
set xlabel "Number of Vehicles in 5 scenarios"
```

```
set ylabel "Time (Sec)"
```

```
plot "sumo_scenario.dat" using 1:3 with linespoints title "Waiting Time (Seconds)",
```

```
"sumo_scenario.dat" using 1:4 with linespoints title "TimeLoss (Seconds)"
```

```
set title "Performance from the SUMO Simulation"
```

```
set xlabel "Number of Vehicles in 5 scenarios"
```

```
set ylabel "Delay (Seconds) and Speed (km/h)"
```

```
plot "sumo_scenario.dat" using 1:2 with linespoints title "Speed in KM/H", "sumo_scenario.dat"
```

```
using 1:5 with linespoints title "Depart Delay (Seconds)"
```

After the above file is written, run it using the command


```
$ gnuplot plot.plt
```

Check the folder for a pdf file created in the name of scenario_sumo.pdf

Number of Vehicles	Jams	Wrong Lane	Yield	Scenario
1543 (S4)	32	1	19	S4
1833 (S5)	2	0	7	S5

Veins + OmnetPP

VANET Scenario in ns2

s5.tcl, mobility.tcl

s4.tcl

s3.tcl

s2.tcl

s1.tcl

VANETs (NS2 Simulator) to generate a trace file (*.tr) and a network animation file *.nam)

To run the tcl file, the command is

```
$ ns file.tcl
```

To run the nam (Network animation), the command is

```
$ nam osm.nam
```

The trace file can be analysed using various scripts...

The screenshot for the animation is provided herewith.

Travelling time (AVg) = Duration (Avg) - Time Loss (Avg)

1 98.48 80.06

2 95.79 89.29

3 97.28 87.24

4 150.15

5 141.37