



TECHNISCHE UNIVERSITÄT
CHEMNITZ

Designing and simulating a Car2X communication system using the example of an intelligent traffic sign

Master's Thesis

to

obtain the academic degree

Master of Science

Department of Computer Science

Professorship of Computer Engineering

Submitted by: Manash Shil

Supervisors: Prof. Dr. Wolfram Hardt

Dipl.-Inf. René Bergelt

Abstract

The thesis with the title “Designing and simulating a Car2X communication system using the example of an intelligent traffic sign” has been done in Chemnitz University of Technology in the faculty of Computer Science. The purpose of this thesis is to define a layered architecture for Infrastructure to Vehicle (I2V) communication and the implementation of a sample intelligent traffic sign (variable speed limit) application for a Car2X communication system. The layered architecture of this thesis is defined based on three related projects. The application is implemented using the defined layered architecture. Considering the availability of hardware, the implementation is done using the network simulator OMNET++. To check the feasibility of the application three scenarios are created and integrated with the application. The evaluation is done based on the result log files of the simulation which show that the achieved results conform with the expected results, except some minor limitations.

KEYWORDS: (Car2X communication, Infrastructure to Vehicle, Intelligent traffic sign, Variable Speed Limit)

Acknowledgement

At this point I would like to thank all the people who helped me to create this master thesis. First I would like to thank Prof. Dr. Wolfram Hardt, Faculty of Computer Science, Chemnitz University of Technology, for giving me the opportunity to do a master thesis at Chemnitz University of Technology with the topic “Designing and simulating a Car2X communication system using the example of an intelligent traffic sign”. It was a great opportunity to work in Car2X technology. This was an interesting topic which helped me to gain knowledge about Car2X communication systems. I also would like to thank Dipl.-Inf. René Bergelt, Researcher in Chemnitz University of Technology, who helped me to finish this master thesis. I would also like to thank Dr. Ariane Heller, Faculty of Computer Science, who helped me to find such an interesting topic.

Table of Contents

Abstract	ii
Acknowledgement	iii
1. Introduction	1
2. State of the Art	2
2.1. COOPERS	2
2.2. CVIS	7
2.3. SAFESPOT	13
2.4. Comparison	21
2.5. Conclusion	22
3. Approach/Algorithm	23
3.1. High Level Architecture	23
3.2. Communication mechanism	23
3.3. Layered Architecture	24
3.4. Reuse of the approach as base	25
4. Implementation	26
4.1. Simulator development vs hardware development	26
4.1.1. Reasons of using simulator	26
4.1.2. Overview of different simulators	26
4.1.3. Conclusion and preferred simulator	29
4.2. DSRC over CALM on the upper layer	29
4.3. Overview of DSRC	30
4.3. Veins Simulation model	34
4.4. SUMO	37
4.5. Development of Layered architecture	40
4.6. Road network and traffic flow	45

4.7. Scenario integration in OMNET++	47
5. Simulation	49
5.1. General scenario simulation	49
5.2. Icy scenario simulation	49
5.3. Accident scenario simulation	50
5.4. Expected results	51
6. Evaluation	52
6.1. Evaluation of general scenario	52
6.2. Evaluation of icy scenario	54
6.3. Evaluation of accident scenario	55
6.4. Evaluation summary	56
6.5. Limitations of overall simulation	56
7. Summary and Prospects	57
8. List of Abbreviations/Definitions	58
9. Table of Figures	59
10. References	60
11. Appendices	63
Appendix A: Passenger car statistics	63
Appendix B: OMNET++ Scenario selection	63
Appendix C: Allocated WSM WAVE Element ID	64

1. Introduction

The number of vehicles on the road is increasing day by day (for statistics, please see Appendix A) which causes more accidents, more traffic jams, etc. To avoid these situations as well as to handle them better, vehicles should get road information (e.g. accident, jam, road surface condition) in a detailed way. The new concept Car2X communication has been introduced to solve these difficulties, where vehicles can communicate with other vehicles or infrastructures also known as V2V (Vehicle to Vehicle) and V2I (Vehicle to Infrastructure) communication.

In Car2X communication information is exchanged through wireless technology between cars and roadside units. The communication is possible in both directions, e.g. Car2Roadside and Roadside2Car. Roadside units can send several information to the cars like traffic congestions, slippery road condition, accident, incident, and road work ahead. Cars can broadcast received information from the roadside unit to other cars within their range. Vice versa, cars also can send received messages from other cars to a roadside unit.

Variable speed limit is a sample application of Roadside2Car communication. In this case, the roadside unit sends speed limit through the wireless medium to the cars within range. The requested speed limit will be displayed directly to the driver's monitor. This is very useful in case of a foggy and heavy rain condition. Because, in such situations a so called Variable Message Sign (VMS) situated on the roadside may not be seen clearly and that can be a safety threat. In this thesis variable speed limit is taken as an example of an intelligent traffic sign application.

This thesis describes and compares three important research projects in the field of Car2X communication in section 2, especially considering the variable speed limit application of these projects. The comparison mainly focuses on the layered architecture, communication mechanisms, advantages, limitations and testing results of these projects. The main goal of this thesis is to define a layered architecture for roadside to vehicle communication (as well as vehicle to roadside communication) and implement a variable speed limit application. The goal is divided into two sub goals: first the definition of a layered architecture for both vehicle and roadside unit, second the implementation of variable speed limit application on the roadside unit. The application variable speed limit calculates the current speed limit in different scenarios (e.g. accident ahead, icy). After research on the related projects, in section 3 the high level architecture is defined in this thesis based on the research result. Afterwards, the implementation, scenario integration and simulations of different scenarios are described in section 4. The section 4 also includes the comparison of different simulators and explains the simulator choice to define layered architecture and implementation of the variable speed limit application. At the end of section 6, the result is evaluated with respect to successful communication between vehicles and the roadside unit (vehicles received and applied the speed limit broadcasted by the roadside unit).

2. State of the Art

The state of the art section describes three different projects related to Car2X communication systems. The research mainly focuses on the roadside to vehicle communication. This section describes the high level architecture, communication mechanisms, test cases and result analysis for each project. It includes a comparison of three projects considering several features like communication medium, communication range, communication speed, advantages and disadvantages. At the end of this section a short conclusion is given based on the comparison.

2.1. COOPERS (Co-operative Systems for Intelligent Road Safety)

The company AustriaTech coordinated the project COOPERS with 39 partners. A part of the budget was contributed by the EU (European Union). The project started on 1st February, 2006 and ended on July, 2010, which took a total of 54 months. The vision of COOPERS project is to increase road safety and better traffic management for a specific road segment via wireless communication between vehicles and infrastructure. [17]

COOPERS based traffic management

Figure 1 shows the high level architecture of the different traffic management systems, including the COOPERS based traffic management system. On the left side of the figure there is the conventional traffic management system where a traffic control center directly communicates with the roadside system using VMS (Variable Message Sign) to show drivers different traffic and road conditions (e.g. speed limit, lane utilization etc.). The middle part of Figure 1 shows the designed high level architecture of the COOPERS project where they have introduced COOPERS Service Centre in between traffic control center, roadside system and vehicle. This chapter mainly focuses on the highlighted middle part. On the right side of the figure the future traffic management system proposed by COOPERS is illustrated where the so called COOPERS Service Centre will be integrated in the traffic control center.

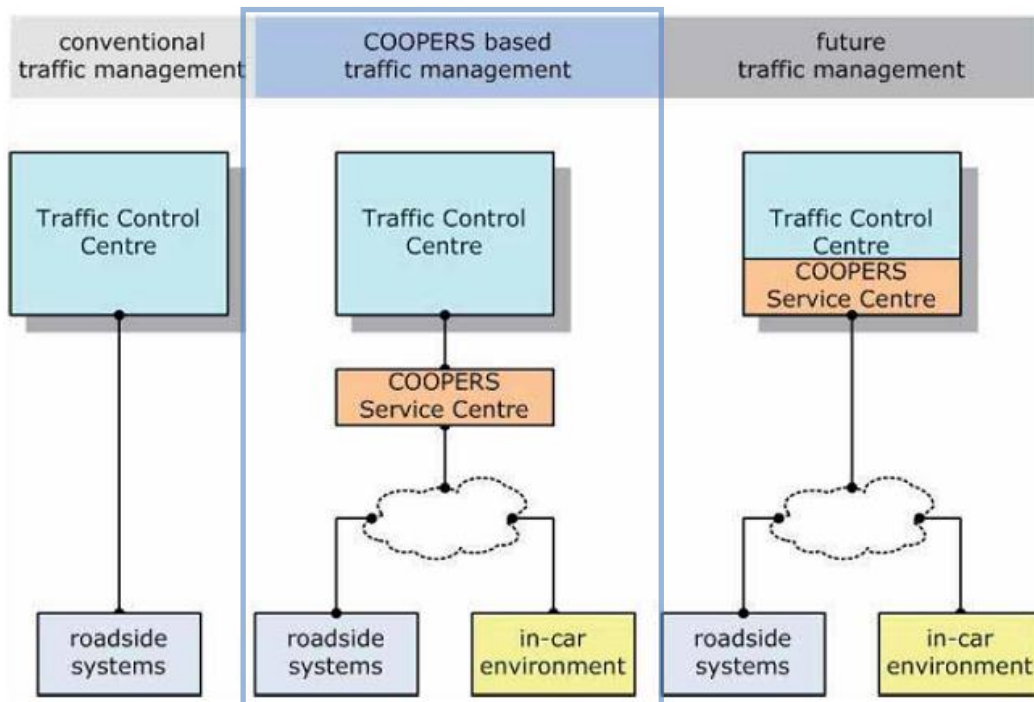


Figure 1: Three models of traffic management system [1]

The following section describes the functionalities of different entities of the above Figure 1.

Traffic Control Centre (TCC)

Source of traffic control related data as well as traffic information like road condition, weather condition etc.

COOPERS Service Centre (CSC)

The COOPERS service center collects traffic information (e.g. road works, traffic congestion, accidents, weather) and traffic control (speed limit, lane utilization) data from the Traffic Information Center and the Traffic Control Center respectively. CSC processes receive traffic related data and broadcast the data to OBU (also called Automotive PC) using TCP/IP via GPRS.

Roadside systems

Equipment installed on the roadside. This system can direct broadcast data acquired from traffic control center or broadcast data after further processing. The intended target is another roadside system and/or vehicles passing by.

Vehicles

The main and most important consumer of the traffic control data. Vehicles can receive traffic related/control data either from the traffic control center or roadside systems or other vehicles. Vice-versa, they can transmit traffic related data acquired by their sensors to the other instances mentioned above.

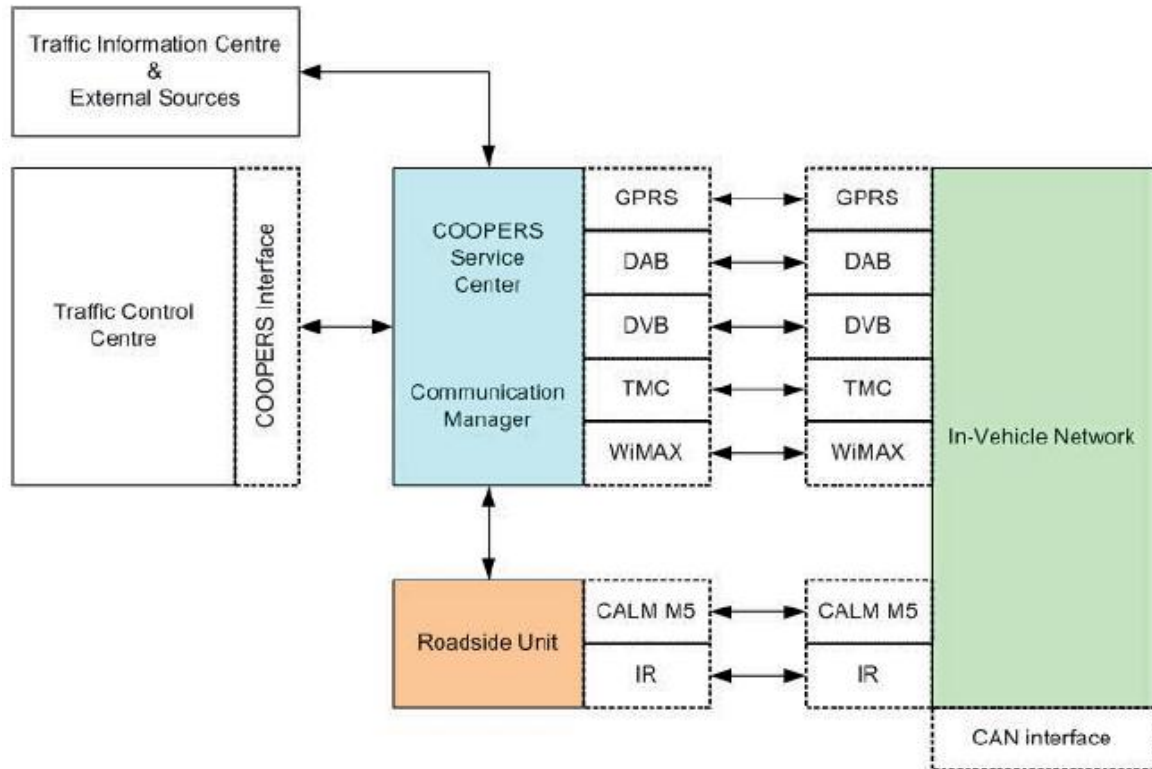


Figure 2: Proposed interfaces between different devices in the COOPERS architecture [2]

Figure 2 illustrates the communication interfaces between the modules. The COOPERS Service Center (CSC) communicates with the Traffic Control Center and Traffic Information Center via Ethernet. After processing traffic related data CSC can directly communicate with vehicles and/or transmit data to RSU for further processing. The direct communication between CSC and vehicle is possible using any technology of GPRS/DAB/DVB/TMC/WiMAX. And the communication between RSU and vehicles uses one of two possible communication technologies, namely CALM-M5 and CALM-IR. As the thesis has taken the variable speed limit application as an example intelligent traffic sign, it focuses on the communication between CSC and vehicles via RSU.

After the implementation CSC created several test sites for the testing of the behavior of the developed application. The following section describes the test site setup.

Test Site

There are a total of four (4) test sites installed for demonstration and testing of the COOPERS services. Test Site-1 lies between the cities Munich, Innsbruck and Trento. It involves 3 roadside operators: ASFINAG (Austria), Autostrada del Brennero (Italy) and OBB (Germany). Other test sites are not considered here as they are not relevant to the topic of this thesis.

Test Site-1 is divided into 3 parts. Test Site-1 D (Germany), Test Site-1 IT (Italy), Test Site-1 ATT (Austria).

Test Site-1 D uses DAB, Site-1 IT and Site-1 AT use CALM-IR as a communication medium.

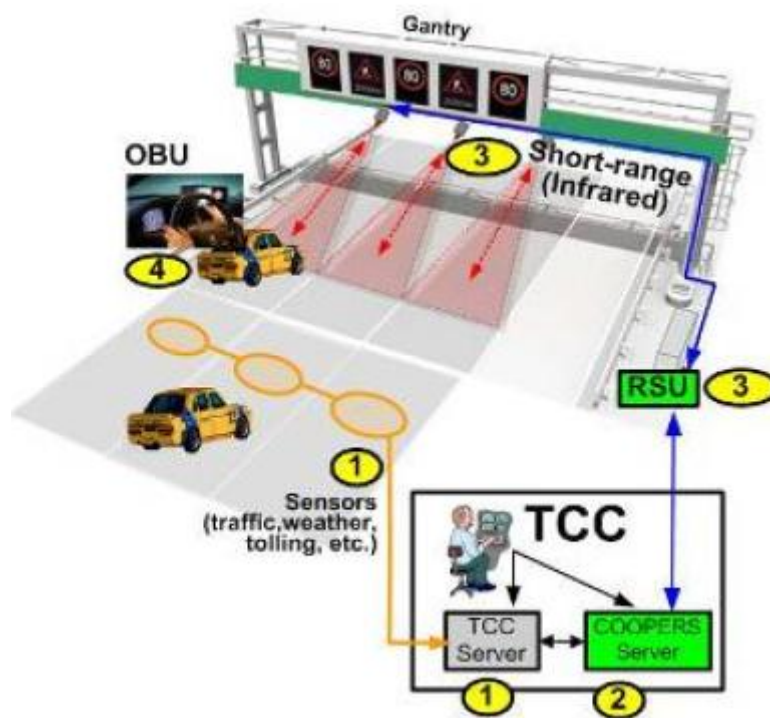


Figure 3: System structure at Site-1 AT/IT [3]

The CALM-IR transceiver is installed on the gantries and controlled by the RSU located also directly on the gantries. RSU is connected with COOPERS Service Centre (located at TCC) through an Ethernet connection. CSC gets the traffic information from TCC server. The CSC server generates COOPERS messages and sends them to the RSU. The RSU sends a corresponding message through its CALM-IR transceiver to the approaching vehicles.

Site-1 D uses DAB for message transmission directly from CSC without using RSU.

Test case

The following table shows the different actions of test case for variable speed limit application [27]

Actions	Expected result
Speed limit information message creation on the roadside	Preparation
Check on APC (Automotive PC) if the message is received and presented to the user, also note the time	Speed limit based on the current road condition and lane is received and presented to the user
Leave motor way and observe whether the speed limit information is lost or not	Speed limit information is lost
Enter the motor way and observe the speed limit information	Speed limit shown on the HMI
Continue driving and note the variable speed limit information period	This is validated after testing the log data
Change the speed limit information message on the TCC and check whether the vehicle received the new variable speed limit information message	Message information is changed
Create termination condition on the TCC	Speed limit information is terminated
Observe vehicle HMI if speed limit information is lost	Speed limit information is lost
Create one of them on the roadside: weather condition warning, traffic congestion warning, incident warning, accident warning, road work information and wrong way driving warning.	Speed limit information message is created after these warnings

Test site result

The test drive was taken for 17km with return drive. It covers 8 gantries, 15 IR-transceivers. All the results are logged. The result is logged for both cases when COOPERS service is ON and OFF. The detailed test results are not publicly available.

It has been observed, when COOPERS service is ON, most drivers follow the message instructions.

2.2. CVIS (Co-operative Vehicle-Infrastructure Systems)

The project is coordinated by ERTICO-ITS Europe and supported by the European Union DG INFSO. The project started on 1st February, 2006 and ended on 30th June, 2010. The aim of the CVIS project was to develop new technology so that vehicles and infrastructures could communicate with each other through various wireless technologies using different wireless mediums (e.g. WLAN, GPRS, 2G/3G) to enhance the road safety. [18][19]

CVIS High Level Architecture

Figure 4 illustrates the high level architecture of CVIS. The CVIS system is divided into three sub-systems (Roadside system, Vehicle system, Central system). The sub-systems are connected via an IPv6 network. Every sub-system has two common modules: CVIS Host and CVIS Router. The CVIS Host manages applications running on the system and the CVIS Router enables the communication with another sub-system.

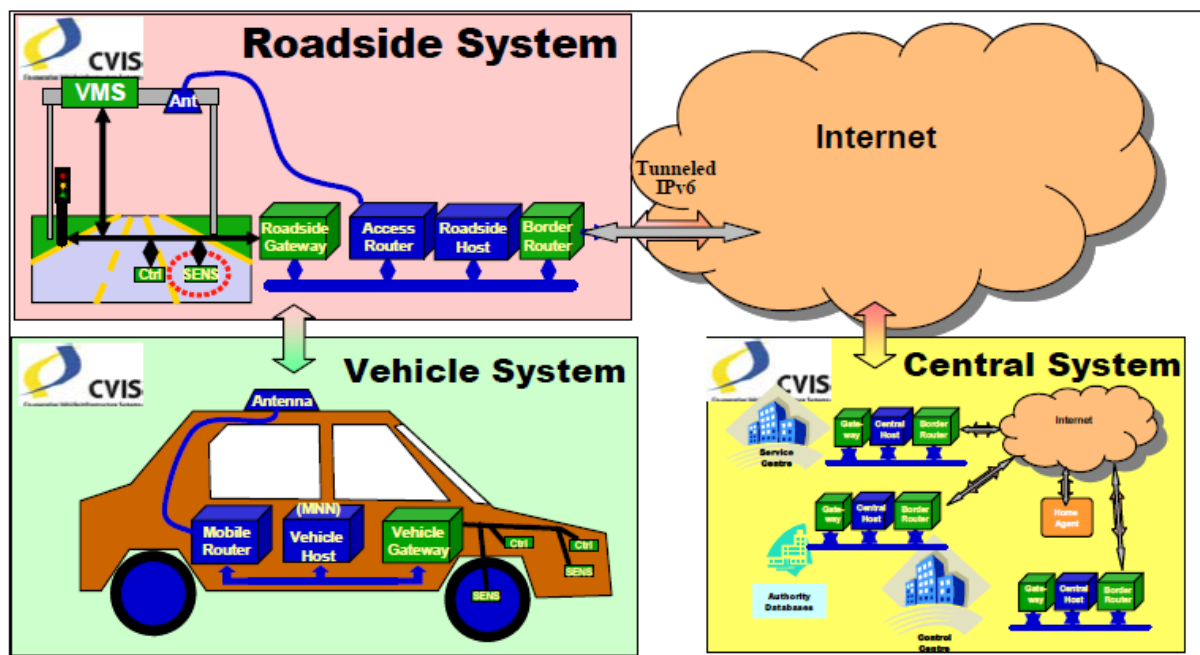


Figure 4: CVIS high level architecture [4]

The following section describes the subsystems and functionalities of the modules inside each sub-system.

The Vehicle Subsystem

The CVIS vehicles are used to monitor the complete traffic conditions on road networks. The vehicle subsystem consists of Vehicle Host, Mobile Router and Vehicle gateway. The Vehicle Host manages services and applications running on the vehicle; also known as mobile node

in the network. The Mobile Router connects a vehicle to the communication network. The Vehicle Gateway is responsible for allowing the services running on the Vehicle Host to access the vehicle data.

The Roadside Subsystem

This sub-system consists of roadside equipment e.g. traffic lights, VMS, antenna. It also has several modules such as Host, Access Router, Border Router and Roadside Gateway. Services and applications are running on the Roadside Host module. The Access Router is responsible for the communication with another sub-system to get the services and to provide the services. The Border Router connects this sub-system to the internet through IPv6. The Roadside Gateway gets the sensor data from the roadside equipment and/or transmits messages to the roadside equipment e.g. traffic lights, VMS.

The Central Subsystem

The central sub-system is a combination of several modules, namely Service Center, Control Center, Home Agent, Authority database and so on to manage service providers, end users, databases, etc. The Service Center manages service applications as well as it constructs and operates service providers. The Control Center is responsible to manage multiple client systems and end users. The Home Agent controls the mobility aspects in the network and works as a message forwarder to the vehicles. Authority Database provides roadside and vehicle data to the service center or control center.

Based on the defined high level architecture CVIS has several applications, e.g. Dangerous Goods, Cooperative Traveler Assistance, Enhanced Driver Awareness, Speed Profile, Cooperative Traffic Control etc.

As the thesis takes Variable Speed Limit as an example of an intelligent traffic sign which is the part of the “Enhanced Driver Awareness” application of CVIS project, only this application is described.

Enhanced Driver Awareness (EDA)

EDA includes several applications, e.g. variable speed limit, traffic jam alert, lane utilization, etc. and The EDA provides two kinds of services: ‘Driving Advice’ and ‘Ghost Driver detection and management’. Current driving conditions on the road network like weather and road condition changes, traffic jam notification, and speed limit are provided by the driving advice service. Ghost driver detection and management service provides detection of ghost driver either by roadside unit or by another driver and necessary actions are taken.

The actors and their responsibilities in EDA

Traffic Manager: This is a human entity responsible to manage the applications running on Traffic Management Centre, how the applications operate under TMC and how the information made available to drivers and travelers. Also the Traffic Manager can decide the use of the road network like lane utilization, set current speed limit etc.

Service Centre Manager: This is a human entity, responsible to manage the Service Centre System. The main objective is deciding how to provide the response of the request from the vehicle drivers by the applications running under Service Centre System.

Personal Service Access Point 1 and 2(PSAP1 and PSAP2) Service Centre Manager: Access Points can be used to report incidents and other emergencies by the travelers. It can also send the information to other travelers. PSAP2 indicates that all the services are not from the same service provider and also enables the data exchange between different instances.

Roadside Controller: This is one or more physical entity situated at the side of the road. These entities display the information like lane use, speed limit, messages etc.

Vehicle Driver: This is a human entity operating the vehicle and communicating with the CVIS entities.

The next section describes the test site design which includes above EDA actors. Several test sites have been installed in different countries and cities. In this thesis only the Germany test site is considered.

Germany test site description and design

There are two test sites installed in Germany, one is an urban test site in Dortmund and the other one is an inter-urban test site in Hessen. Both test sites focus on the Cooperative Monitoring (COMO) technology and Cooperative Urban (CURB). In Dortmund 3 vehicular units and 4 roadside units are installed. On the other side the Hessen test site includes 4 vehicular units and 10 roadside units.

Objectives of test site Dortmund

This test site focuses on COMO technology and also strategic routing developed in the CURB sub project. The main objectives include implementation, integration, testing and evaluation of urban applications (e.g. green light priority at intersections, smart routing) and COMO technologies (e.g. data fusion, traffic computation, network monitoring). Another important goal is to demonstrate the solution in a real environment to check its feasibility and benefits.

Objectives of test site Hessen

This test site focuses only on COMO technology. The main objectives include implementation, integration, test and evaluation of the COMO technologies: FCD (Floating Car Data) events and EDA (e.g. speed limit alert, ghost driver detection).

The following section illustrates the test site description, necessary equipment and data exchange between the different modules. As EDA is tested only in test site Hessen, only this test site is considered here.

Hessen

The test site Hessen is approximately 11km long and both directions are considered. There are 5 gantries and each has a distance of 1000m. The test site has installed 5 RSU on each direction.

Vehicles are equipped with a CALM-M5 module, antenna, Ethernet card, PC-kit CVIS-S-R-2-018 and a cellular communication module. The Car-PC is connected to the HMI to display the information of the EDA. The Car-PC is connected to the CVIS-Host via ethernet where CVIS applications are running. External communication with the RSU and the control center is done by the CVIS Router and antenna.

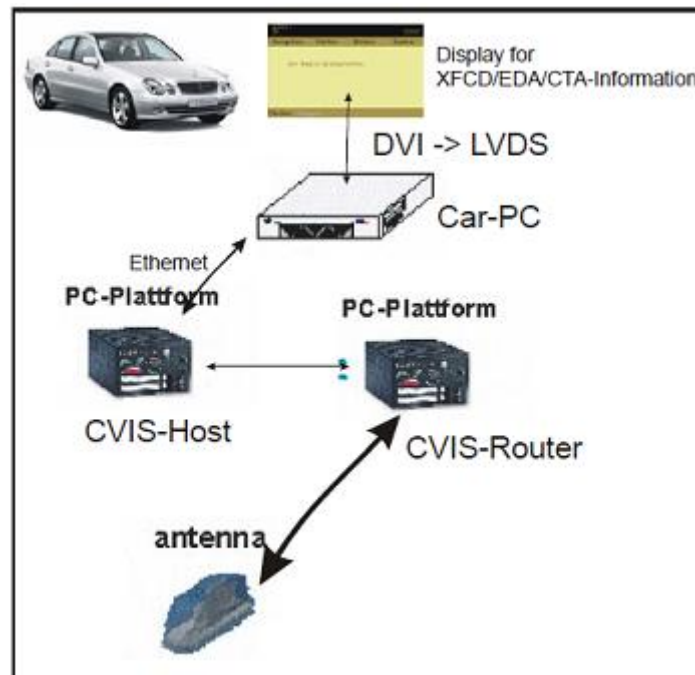


Figure 5: Hardware configuration of a test vehicle [5]

The road side is equipped with two PC platforms (one acts as CVIS Host and another one as CVIS Router), CALM-M5 module, antenna, cellular communication module and external border router. Two antennas are used for a single RSU to cover both directions.

The following figure 6 illustrates the communication between different modules in test site Hessen. The CVIS/COMO Center lies between Traffic Management Center (TMC) Hessen and vehicle/roadside system. The CVIS Center exchanges data with TMC via IPv4 web service. Data exchange includes traffic congestion, traffic speed, speed limit, overtaking prohibitions, etc. Through IPv5 the information are exchanged between CVIS Centre, RSU and vehicles. The communication between RSU and vehicle is handled by CALM technology.

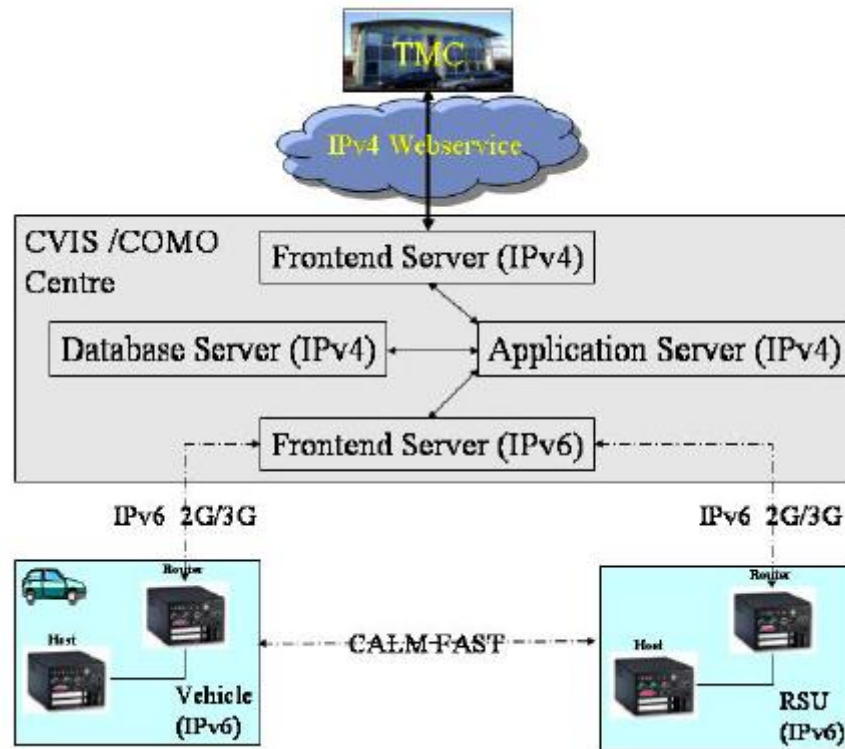


Figure 6: Architecture of the interconnections between vehicle, RSU, COMO Centre and Traffic management Centre (VZH)
[6]

Germany test site result

COMO EDA speed limit advice in test site Hessen

The Traffic Management Centre Hessen advises speed limit to the CVIS Centre via DATEX II interface. The CVIS Centre sends the received message to the RSU. The RSU broadcast the speed information to the CVIS equipped vehicle.

The Traffic Management Center assesses the current road conditions and because of the heavy traffic density it decides to decrease the speed limit temporarily to 40 km/h. The new speed limit is transmitted to the RSU and which starts broadcast the new speed limit. Equipped vehicles in the range receive the new speed limit and the driver is informed through VMS. Figure 7 shows the resulting communication flow from TMC to vehicle.

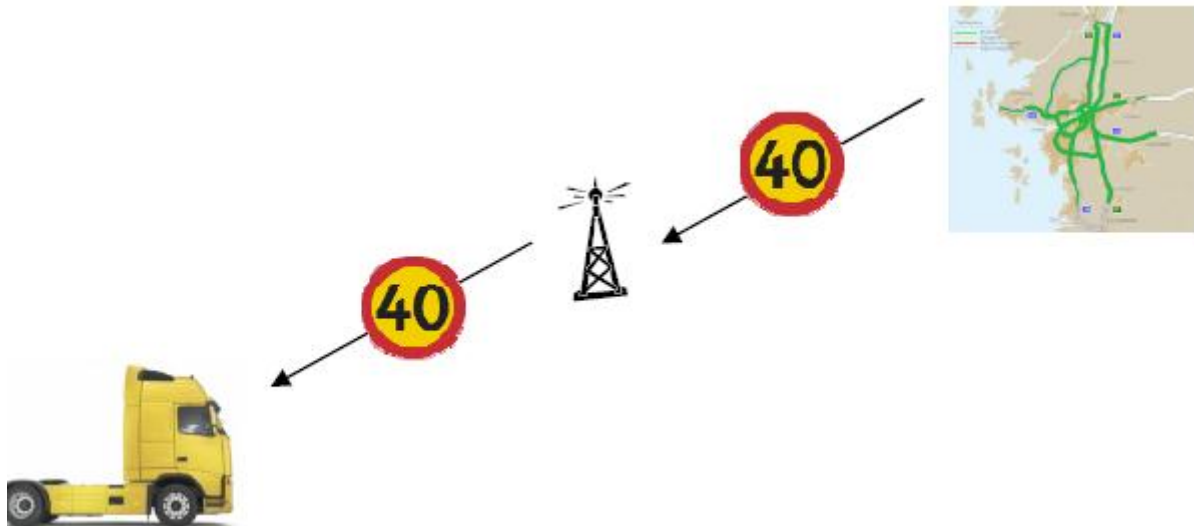


Figure 7: Speed limit [7]

2.3. SAFESPOT

The SAFESPOT project was coordinated by Roberto Brignolo Centro Ricerche FIAT (IT) and included 52 partners. It was co-funded by the European Commission Information Society and Media in the 6th Framework Program. The project started in February, 2006 and ended in January, 2010. The aim of the project was to design a cooperative systems for road safety based on the wireless communication between vehicles and infrastructures. [20]

SAFESPOT has infrastructure based and vehicle based subprojects. In the infrastructure based project, vehicles communicate with infrastructure for information like traffic jam alert, speed limit, etc. The vehicle based project involves the communication between vehicles. Variable Speed Limit which is the example of an intelligent traffic sign application for this thesis is part of the infrastructure based sub project “COSSIB - Cooperative safety systems infrastructure based” named Speed Alert (SpA) in SAFESPOT. And hence, this project overview only includes necessary information for SpA application. Before starting the description of the target SPA application, at the beginning high level VANET architecture and system components are described. Then the tests and their results come at the end.

SAFESPOT VANET architecture

The applications are divided in two parts: HMI part and Cooperative part as shown in the Figure 8. The Cooperative part is responsible to generate messages for drivers and HMI part is responsible to inform a particular driver.

Acquired sensor data are transmitted to LDM (Local Dynamic Map) after refinement. After scenario analysis applications decide which information is useful for other nodes. Then this information is sent out as a Cooperative Awareness (CA) message also called Beacon generated by the Message Generation module. The generated CA message is sent to the vehicle. The header of a CA message is called NL Beacon which contains the node information like node identifier, timestamp, position, heading and velocity. CA messages also contain node status and node environment as payload. The CA header and timestamp is the responsibility of the VANET module and payload is the responsibility of Message Generation module.

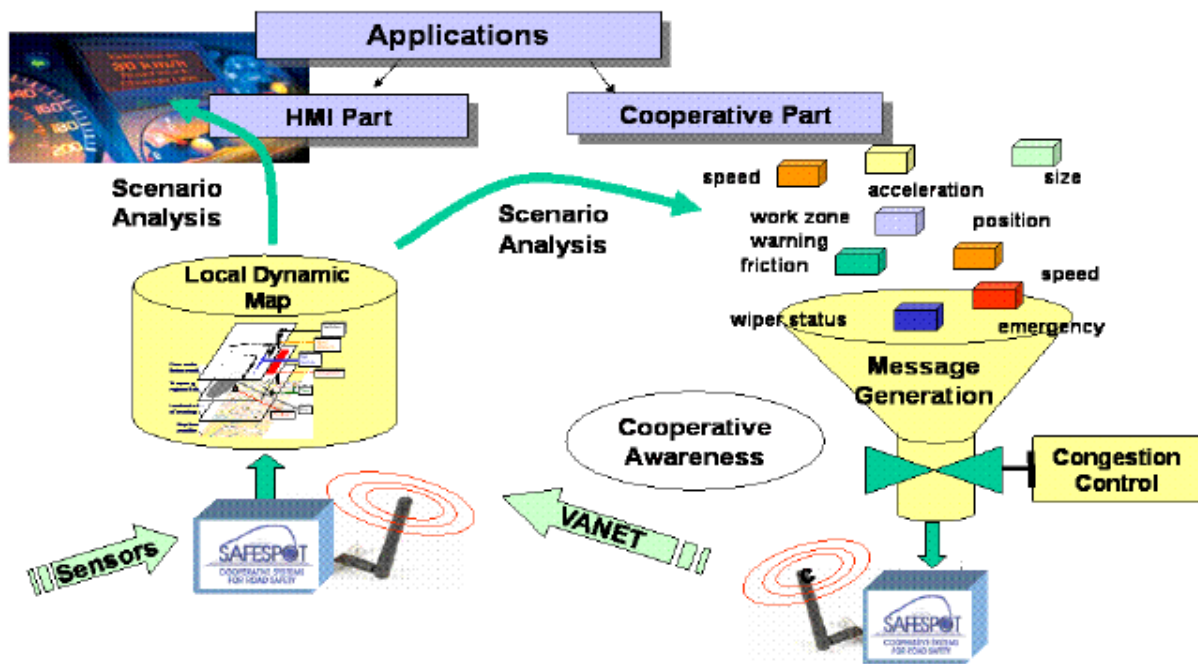


Figure 8: VANET architecture (LDM and Message Generation) [8]

System Components

There are several SAFESPOT components including Roadside sensors, Roadside alerts, Nodes, RSU (also called MFO: Multi-functional Outstation), Centre (TMC/TIC), Local Area Network (LAN), Wide Area Network (WAN), Equipped vehicles, Non-equipped vehicles. Figure 9 shows all the components and their position on the road network.

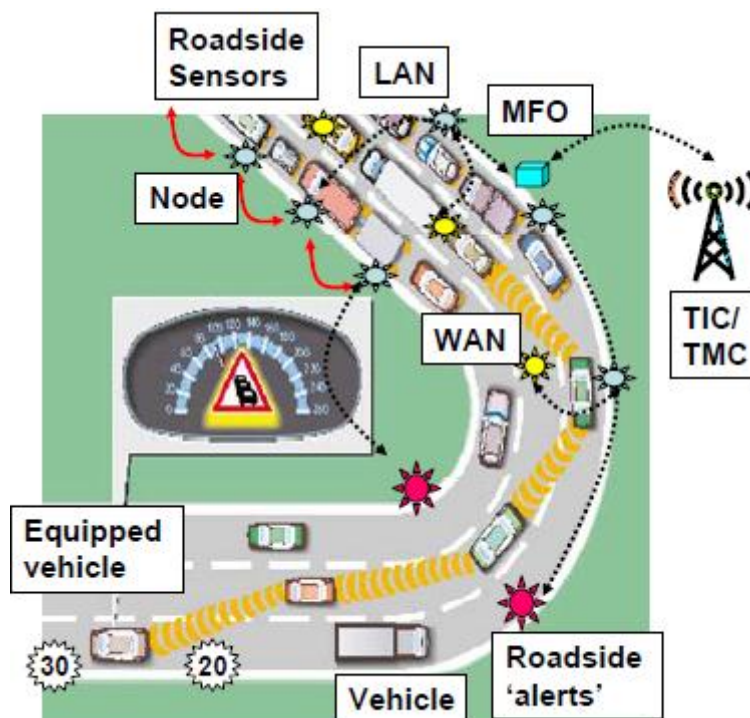


Figure 9: Infrastructure system components [9]

The following section describes the functionalities of different system components.

Roadside sensors

Physical device installed on the roadside. The sensing system consists of several technologies like radar, infrared, laser, video, inductive loop, ultrasonic, etc. Sensors are used to detect events, conditions or safety critical situations, e.g. accidents, wet road surface, foggy weather, sharp curve, wrong way driving etc.

Roadside Alerts

This is a physical device installed also on the roadside. The responsibility of this entity is to give the warnings or messages to the road users as visual or other signal. As an example, it advises recommended speed limit, lane utilization, and obstacle ahead.

Probe Vehicle

This is a physical entity which acts as a mobile sensor. It collects the data of surroundings and makes it available for the roadside unit directly or via a node. Non-equipped vehicles are treated as part of the traffic population on the road network.

Node

It is a physical entity installed on the roadside. In some cases it can be the part of the sensing system. A node consists of several modules: sensing module, processing module, communication module. It receives signals from the sensing devices, conducts data filtering, processing, diagnostics and then transfers the data to the roadside unit. Some nodes can be able to exchange data with vehicles and send commands to the warning system.

Roadside Unit/Multi-functional Outstation (MFO)

A physical entity makes the further data processing received from the nodes. Based on the data, applications running on the system generate safety-critical warning messages for the road users. It contains the necessary digital maps, database, diagnostic functions required by the applications. It communicates with the nodes, other RSUs and Centre (TMC/TIC).

Centre

This is a non-roadside equipment located in the Centre such as Traffic Management Centre (TMC) or Traffic Information Centre (TIC) where some applications can be installed. It permits the system a remote view of an area and gives access to the data of a wider area if necessary.

Local Area Network

This is a communication network used to exchange data between nodes and communicate with a roadside unit within the local area. Depending on the type of application and communication system used it covers 250 to 1000 meters.

Wide Area Network

This a communication network for the roadside units or group of units to exchange data within a wide geographical area.

Application: “Speed Alert (SPA)”

The Speed Alert application is divided into three parts: legal speed limit, calculation of recommended speed based on the dynamic events on the road and definition a speed profile according static black spots. All processing is done in the road side unit based on the data stored in the LDM (collecting data from the vehicles and infrastructure sensors). If a new driving rule is generated by the Speed Alert application, the information are sent to the drivers via ad-hoc network for the vehicles in the range, via VMS for non-SAFESPOT equipped vehicles. Information are also stored in the LDM for the oncoming vehicles in the long range. When the application starts, at first it receives data from LDM. Based on the received LDM data the application defines the speed limit or speed profile depending on the current situation. At the end, a relevant message is generated and sent.

Test and result of Speed Alert (SpA) application

The application gives drivers the recommended speed based on road surface condition, weather condition, traffic flow on the road, road works, jam, etc. The ‘Speed Alert’ application is divided into three sub applications: ‘Legal Speed Limit (SpA_01)’ which warns the driver about the legal speed of the road, ‘Critical speed warning (SpA_02)’ warning driver about the new speed based on the environment and traffic condition and ‘Excessive Speed Alert (SpA_03)’ which defines a new speed limit based on the road definition and status.

Test case: Speed transition at a straight road for SpA_01

The goal is to prove the functionality of the sub application SpA_01 at a straight road. In the test scenario a vehicle enters in RSU range and sends a beacon to RSU. As soon as the RSU received the beacon it detects the vehicle. The RSU transmits legal speed limit to the vehicle if the speed of the vehicle is higher than legal speed. The received message is displayed to the driver through HMI.

The test conditions for the test cases are considered that based on simple classes the vehicle types are identified by RSU, nominal speed ranges are set to 90km/h, 110km/h and 130km/h for the vehicles. In the middle of the road vehicle speed is decreased to 50km/h, 90km/h and 110km/h respectively of nominal speed (by considering the speed limit advice by the RSU).

The expected results of this test case are that the RSU detects the vehicle type based on the beacon message sent from the vehicle. The vehicle is warned by RSU if the vehicle speed is greater than the legal speed limit within 1 sec of entering in the area covered by RSU.

Obtained values / Results (test site result): The following table shows the test results of the above test case. [22]

	Result	Comment
Message management: System shall be able to decide the receiver of the message	OK	
Priority message: System shall be able to prioritize the transmitted message	OK	
Data exchange: System shall be able to send data to the several supporters(e.g. VMS, Radio)	OK	Only VMS was verified
Data query from LDM	Partially OK	NavtTeq LDM and subscription mechanism was used. LDM crash while closing the subscription
Data receive from LDM	OK	
Simultaneity: System shall be able to handle all vehicles simultaneously in the covered area	OK	
Scalability: The performance for the system shall be the same for 2 or 80 vehicles	Partially OK	Only 3 vehicles
Time of Warning Generation: System shall be able to generate and transmit warnings on time	Partially OK	Sometimes has huge latency(higher than 1s)
Robustness of System	Not tested	Hardware was not tested under this condition
Communication range: At the critical points short range communication is available (minimum 600m)	Not OK	Ranges were around 300m
Simultaneous Communication: In the vicinity system shall be able to communicate with all vehicles. Minimum radius $\leq 600m$	Not OK	Ranges were around 300m
Environmental data: Weather data(e.g. rain, fog) shall be received from the sensors	OK	
Vehicle position and speed: Detects vehicle position and speed	OK	
Warning transmission: Warning / recommendation transmission to vehicles	OK	
Road status: System shall receive data on the critical conditions (e.g. obstacle / accident, wet road surface)	Partially OK	Mainly handled the weather related situations

Test case: Speed transition at a straight road with different weather conditions for SpA_01

The goal of the test is to show that this sub application can handle the weather conditions (e.g. rainy, foggy) and the associated speed limit. This test case covers complex road networks (traffics initially have excessive speed) and road types (interurban, urban and highway). The scenario is the same as with test case 1.

Test Conditions include vehicles entering on RSU range which have a defined road trip; considered speed in the range of 50km/h to 130km/h; application should adjust the new speed limit based on the weather conditions (e.g. rainy, foggy) and send a beacon to vehicles.

The results expected for this test case are as follows: application detects weather condition and sets new speed limit. In case of excessive speed compared to the new calculated speed limit vehicles are warned by beacon message.

Test case: Evaluation of safety distance and associated speed for SpA_02

The goal of this test is to define a safe area and associated speed according to the warning given by the hazards and incident warnings. The test defines three different areas and associative speed before entering the incident area. Figure 10 shows the incidence area (marked as black) and three different areas (marked as red, yellow and green) before the incident area.

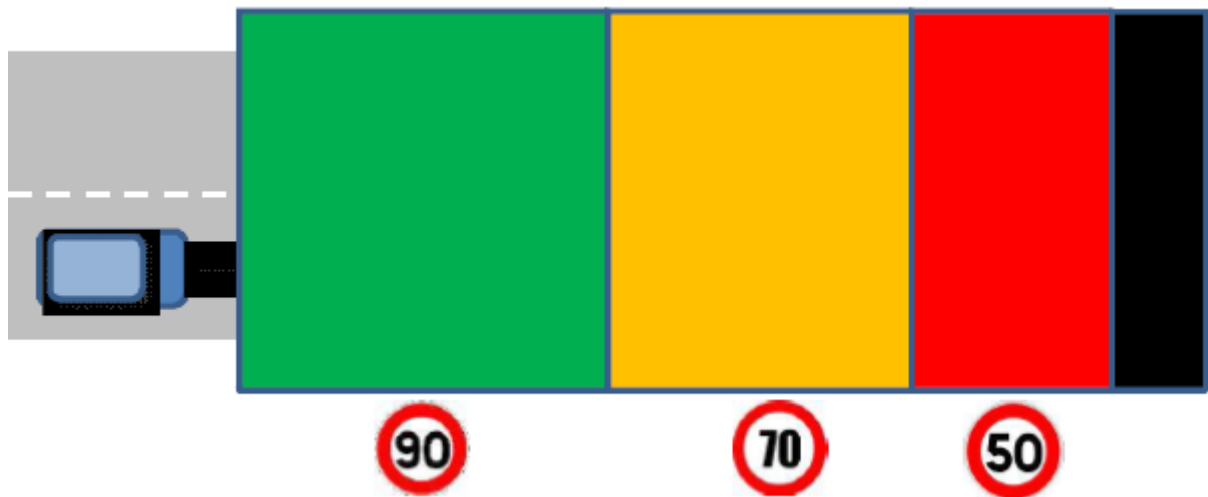


Figure 10: Three areas and ahead incident area (black) [10]

Test conditions for this test case include that the application can define the areas for different speed limits and defines the entrance speed of these areas; also defines the impact area where to give the final speed warning.

The result expected is that road areas and associated speed are defined by the application and the application also defines the impact area.

Test case: Generation of safe area under several H&IW events for SpA_02

The goal of this test is to show the behavior of the application against multiple incidents on the road network.

Considered test conditions are multiple hazard and incidents warning, the generation of events and that the application handles multiple speed warning.

The expected result is that the application manages several incidents in the same road section and defines the safe area.

Test case: Basic validation of SpA_03

The aim of this test is to define a speed profile based on road geometry and road condition.

Test conditions include: vehicle entering on the RSU range sends beacon to RSU; based on the beacon message from vehicle RSU detects vehicle type; RSU transmits warning message to the vehicle if the vehicle speed is higher than the defined speed limit and the warning is shown to the driver.

The expected result of this test case is that in case of excessive speed the driver is warned by the RSU, otherwise the vehicle OBU is concerned about the legal speed limit.

Obtained Values/Results (test site result): The following table shows the test results of the above test case. [23]

	Result	Comment
Directly address a Vehicle: The RSU shall be able to transmit data directly to one vehicle or broadcast to all vehicles	OK	
Time of data delay: In the roadside the delay between message generation and transmission to vehicle should be less than 50ms	Not OK	A lot of delay
Connection time: Connection time between approaching vehicle and RSU should be less than 0.8s	Not tested	Only beaconing message was available on the log
Static Map contents: LDM shall describe in details and accurately the road geometry in critical intersections	OK	
Speed and position of vehicles: RSU shall be able to receive current vehicle speed and position	OK	
Warning transmission: RSU shall be able to transmit warning to the vehicle	OK	

Test case: Road surface condition interaction with SpA for SpA_03

The goal of this test case is to verify that the application can handle environmental condition and thus defines the associated speed. The road surface is also considered like for test case 1.

Different weather conditions are expected for this test case. The expected result is the application advises a different speed limit for different weather conditions and also that an excessive speed warning is given to the driver if the speed is higher than the calculated speed limit.

2.4. Comparison

	<i>COOPERS</i>	<i>CVIS</i>	<i>SAFESPOT</i>
<i>Medium</i>	Infrared, Wireless	Wireless	Wireless
<i>Communication Technology</i>	CALM-IR, DAB	CALM-M5	VANET(IEEE 802.11p)
<i>Communication Range</i>	CALM-IR: Typically 30m DAB: around 200m	Several 100 meters	10-300 meters
<i>Communication Speed</i>	CALM-IR: Net data rate- 1.2 Mbps DAB: Net data rate- 1.5 Mbps	Data rate: ~6 Mbps	Data rate: ~6 Mbps
<i>Advantages</i>	CALM-IR: short connection set up time(~10ms), good data throughput, does not need any license DAB: Improved end-user features, variable bandwidth, reception quality	1. supports directional and Omni-directional communication 2. Regional DSRC cooperation 3. Multiple radios/interfaces/antenna management 4. GPRS/UMTS network interconnectivity	1. Recommended speed calculated on the roadside unit based on the data received from sensors. So, no dependency on TCC
<i>Disadvantages</i>	CALM-IR: short range DAB: coverage, transmission cost, compatibility, power requirements		1. Expensive because of installation of many sensors, alert systems, RSU

2.5. Conclusion

All three projects communicate with vehicles from Roadside Units through wireless technology. The architecture of COOPERS based traffic system and CVIS is much simpler than SAFESPOT. In the SAFESPOT implementation traffic related data are mainly fetched from the several sensors installed on the roadside whereas the other two systems acquire traffic related data mainly from the Traffic Control Centre (TCC). The benefit of SAFESPOT in this case is that there is no dependency on the required data. As all traffic related data are available at TCC, this is a redundant approach to acquire traffic related data. Considering the most interesting part of this project, COOPERS has only tested communication via CALM-IR and DAB where CALM-IR covers only short range and DAB is not widely available. Best suited for communication is CALM-M5 used in the CVIS project on top of IEEE802.11p; SAFESPOT also uses the IEEE802.11p protocol for wireless data transmission. The Data rate (~6Mbps) of the CVIS and SAFESPOT is much higher than that of COOPERS.

3. Approach/Algorithm

3.1. High Level Architecture

Based on the overview of the different related projects, Figure 11 depicts the architecture of this project. With respect to the sensors installed on the roadside, the architecture of COOPERS and CVIS described in section 2.1 and 2.2 respectively is better suited for the application than SAFESPOT. Therefore, developed architecture for this thesis is based on COOPERS and CVIS architecture (both CVIS and COOPERS have the same high level architecture for the Roadside to Vehicle Communication). A chosen communication mechanism for data transmission from RSU to Vehicle is based on the CVIS and SAFESPOT projects. The following section defines the functionalities of several entities of the defined high level architecture. The entities are: Traffic Control Centre (TCC), Roadside Unit (RSU) and Vehicle.

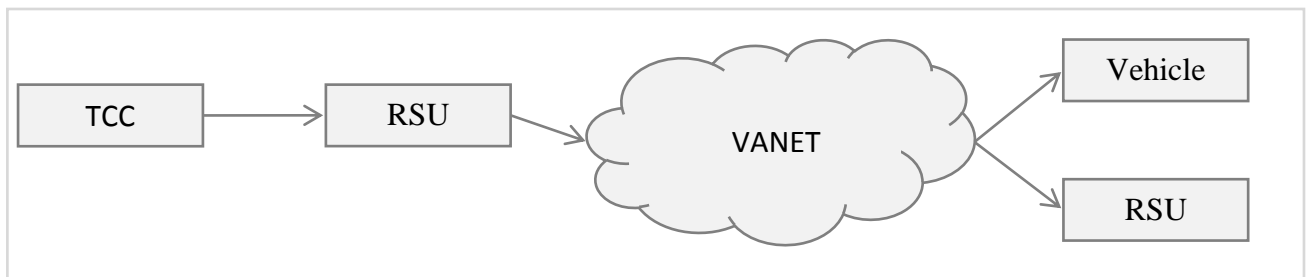


Figure 11: High Level Architecture

Traffic Control Centre provides the traffic related data to the Roadside Unit. After processing the data the application will decide the critical speed limit and will transmit it to the Vehicle and/or other RSU. The received speed limit will be shown to the driver through HMI. The application will send the speed limit also to the VMS for non-equipped vehicles. The RSU is basically a combination of two elements: Service provider (containing speed limit application) and wireless transmitter.

3.2. Communication mechanism

The communication between TCC and RSU will be done via ethernet connection if there is short distance between TCC and RSU. Another possibility is that RSU and TCC communicate using IPv4. The number of different services (e.g. weather conditions, incidents, etc.) running in TCC provides different kinds of data required by RSU or vehicle or other components. The application running on the RSU acquires necessary data of the road network (e.g. traffic congestion). Based on the received data the RSU calculates the speed limit for the specific road and through IPv6 sends a beacon to the vehicles.

3.3. Layered Architecture

Figure 12 illustrates the layered architecture designed for both vehicle and roadside unit. The designed layered architecture uses IEEE802.11p in the physical layer which is an extension of IEEE802.11a protocol. IEEE802.11p has been especially introduced to support VANET (Vehicular Adhoc Network). On the upper layer the designed architecture uses CALM technology.

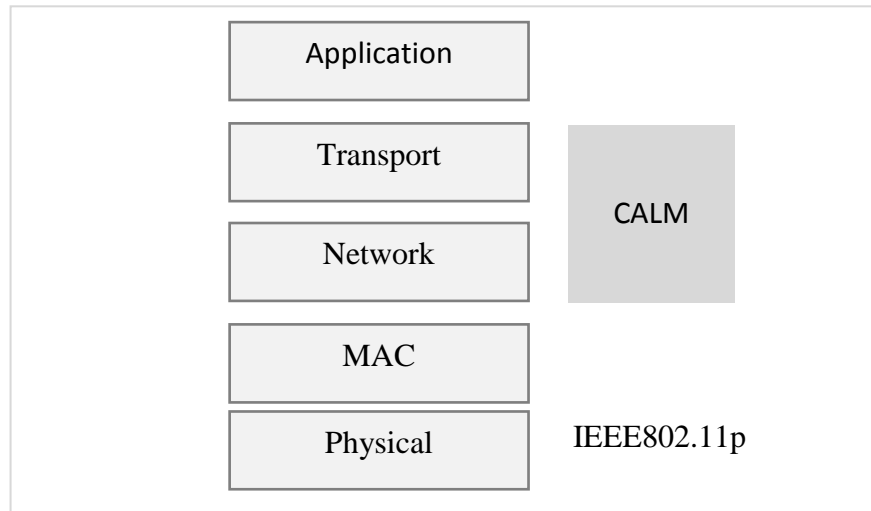


Figure 12: Layered Architecture concept

CALM (Communications, Air-interface, Long and Medium range)

The CALM concept provides a layered solution for continuous and quasi-continuous communication between infrastructure and vehicles or between vehicles. The communication is done by wireless telecommunication media available in a particular area, and also can be migrated to other available media if required. The communication is based on IPv6.

There are several benefits of using CALM. It combines several communication media (WAVE, 2G/3G, GPS). It is possible to implement several communication media on the same architecture. In that case it can communicate with the media which is available in a particular area. The main advantage of several media is if one media fails, the communication is still possible with another implemented media (if there are more than one media implemented then it is necessary to define primary media and secondary media). Figure 13 illustrates the classic architecture of the CALM technology. At the physical layer it is possible to define different protocols for different kinds of communication. All the media use IPv6 on the network layer and corresponding transport protocols (e.g. TCP, UDP, etc.) are used based on the requirement. CALM supports several media such as ISO21214InfraRed, ISO21215Ghz (Similar to DSRC), ISO21216Millimetre, ISO21212Cellular2G, ISO21213Cellular3G, ISOxxxxHC-SDMA, IEEE802.20, ISOxxxxxSatellite, ISO15625DSRCProfiles, EN12834DSRCL7 and local initiatives conforming to ISO15628.

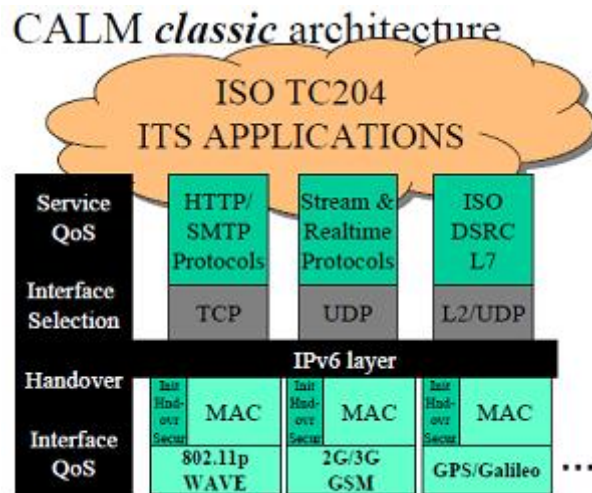


Figure 13: CALM classic architecture [11]

3.4. Reuse of the approach as base

The architecture and communication mechanism is more generic and can be used for more than the speed limit application which will be implemented in this thesis. Therefore, other related infrastructure based applications can be implemented considering this project as a base.

The proposed architecture can be reused as the base of other intelligent traffic sign applications, especially for the “Infrastructure to Vehicle” communication because of using the specified communication mechanisms and protocols on the different communication layers (PHY: IEEE802.11p; Transport: TCP).

4. Implementation

4.1. Simulator development vs hardware development

The algorithm will be developed on a simulator. In that case it is necessary to design the whole network, define communication mechanisms and required protocols on the simulator environment then evaluate the result of the simulation.

4.1.1. Reasons of using simulator

The decided approach uses VANET (IEEE802.11p). Current hardware setups do not have this protocol installed. However, it is possible to implement the approach with current setups (without using IEEE802.11p but using IEEE802.11a/b/g/n). In that case, hardware development will be a bit different than the actual approach defined here. That is why it is better to develop the approach on a simulator where full communications can be defined and tested. Another reason is, the real tests require a high number of participating vehicles which is not available. But using a simulation approach this can easily be satisfied. As well as for the restriction of law, real life traffic evaluation is not possible.

4.1.2. Overview of different simulators

As it is decided to develop the approach on a simulator, now it should be decided on which simulator it will be developed. There exists a great variety of simulators for distributed systems and networks. After studies on different simulators it is easier to decide which simulator best fits. In the following, different simulators are discussed:

OMNET++

OMNET++ is a simulation environment emphasizing on simulation of communication networks. It provides a component architecture for models. Components also known as modules are programmed in C++, then integrated into models through NED (Network Description language). It has rich GUI support. Different simulation models can be integrated into OMNET++ for special tasks (e.g. ad-hoc networks, wireless sensor networks).

Components

- Eclipse based IDE
- Simulation kernel library
- NED network topology description language
- Simulation execution from GUI(Tknev)
- Simulation execution from command line environment(Cmdnev)

Platform

OMNET++ and OMNET++ IDE runs on Windows, Linux or Mac

Simulation Process: Figure 14 shows the sequential working process in OMNET++.

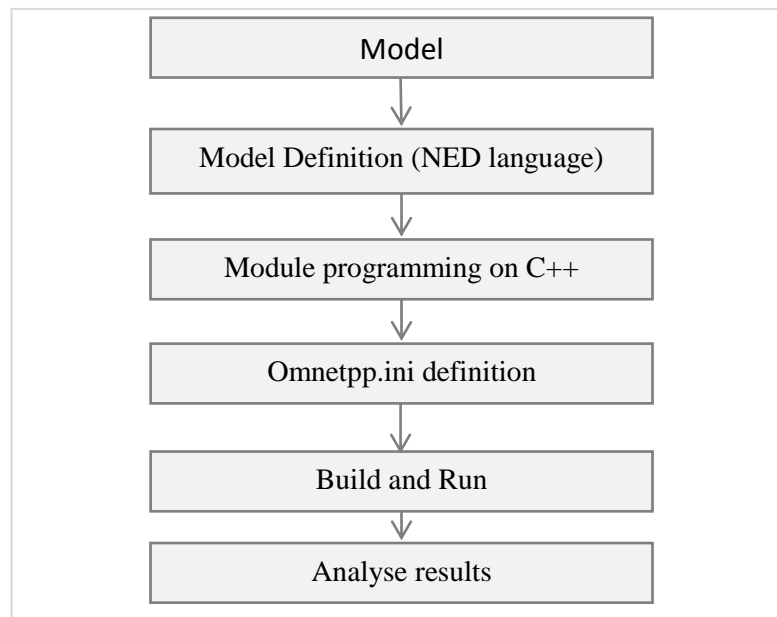


Figure 14: OMNET++ simulation process

Model: The OMNET++ model combines components or modules which exchange messages between them. Modules can be compound modules where several modules are grouped into one large module.

Model Definition: Define the model structure in high level NED (Network Description) language.

Module Programming: Modules should be programmed in C++ using simulation and kernel library.

Build and Run: Links kernel and necessary library files then builds and runs the program.

Analyze results: Simulation results are stored as output vector and output scalar files. Results can be visualized through the tool provided by the IDE. Results can also be processed with Matlab or other tools.

Advantages

- Well known eclipse based development
- Rich GUI for development
- Integrates various simulation models and frameworks(wireless sensors network, ad-hoc network) according to the requirements in the project
- UML modeling, bug tracker integration, database access is possible
- Portability: generated result can be analyzed with other tools like Matlab, R.

- Rich result analysis tool “Scave” to analyse the generated result. A dataset can be defined; possible to show various types of output(Line chart, Bar chart)

Limitations

- No functionality for multiple radio interfaces
- Limited features for parallel simulation(blocking event synchronization)

SimANet

SimANet is a simulation environment for mobile network communication and mainly focuses on mobile ad-hoc networks and sensor networks. It supports the concurrent operation of multiple radio modules within same network.

Advantages

- Easy extension with additional modules like power consumption, communication complexity, barrier simulation
- Provides both GUI and command line mode for simulation
- Parallel execution of nodes is possible
- High simulation performance

Limitations

- As the simulation networks is communication dominant, efficient parallelization is difficult.
- Simulating nodes in critical sectors slows down the simulation speed
- Time synchronization problem on different simulator instances

NS-3(Network Simulator 3)

NS-3 is a discrete event network simulator. The intention of NS-3 is to develop an open simulation environment for networking research.

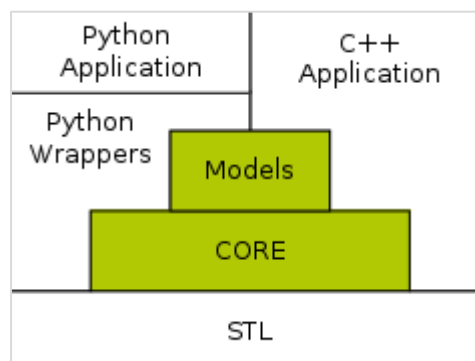


Figure 15: Architecture of NS-3 simulator [12]

It contains network simulation models as C++ objects which are wrapped in python. One can develop the application either in C++ or python.

Advantages

- Different real-time simulation models can be developed and integrated into the NS-3 framework
- Reuse of real application and kernel code

Limitations

- No IDE is integrated
- Few simulation models
- Only works under linux platform

4.1.3. Conclusion and preferred simulator

OMNET++ has more rich simulation models than the other mentioned simulators. Considering this project, OMNET++ has MIXIM simulation model for ad-hoc network, wireless sensor network, vehicular network. OMNET++ and SimANet have an integrated IDE but NS-3 does not. OMNET++ has a rich analysis tool for the output result compared to the two alternatives. NS-3 only supports linux development whereas OMNET++ runs on windows, linux and macos. SimANet is better than the other simulators with regard to parallelization. OMNET++ and NS-3 have more complete rich online help than SIMANET. Before NS-3 there was NS-2. However, NS-3 is not backward compatible considering NS-2 but OMNET++ is backward compatible. There is no clear specification of SIMANET about backward compatibility.

Based on the evaluation, OMNET++ has been chosen for the development of this thesis.

4.2. DSRC over CALM on the upper layer

DSRC and CALM both use 5.9GHz frequency. CALM supports several media whereas DSRC does not. To develop ITS applications CALM and DSRC both use IEEE802.11p on the lower layer. CALM uses IPv6 on the network layer where DSRC has two options: for safety applications DSRC uses WAVE Short Message protocol which lies in the transport and network layer, for non-safety application it uses TCP/UDP in the transport layer and IPv6 on the network layer. Like CALM technology DSRC does not support several media (e.g. 2G/3G, GPS, etc.) installed at the same time on a single device, which is an undoubted advantage of CALM technology. But CALM is still in development and hence all the specifications like how to manage the media and different layers are not available. Though the projects we have discussed in section 2 (COOPERS, CVIS and SAFESPOT) are using CALM technology, there is no clear specification how they implemented it. For this reason, the implementation is done using DSRC technology. Since the variable speed limit application is considered a safety application, the implementation is done using WSMP on the upper layer.

4.3. Overview of DSRC (Dedicated Short Range Communication)

Figure 16 shows the protocol stack used for DSRC communication, including several standards on different layers. At the physical and mac layers it introduces IEEE802.11p WAVE (Wireless Access for Vehicular Environment). In the middle layer it uses several IEEE1609 standards: 1609.4 for channel switching, 1609.3 for network services which includes WSM protocol and 1609.2 for security services. It also supports TCP/UDP in the transport layer and IPv6 on the network layer. Depending on the requirements of the application either WSMP or TCP/UDP IPv6 is used. Generally, all the safety applications use WSMP and non-safety applications use TCP/UDP IPv6 on the transport and network layer. SAE J2735 is a basic message set dictionary, especially used for V2V communication.

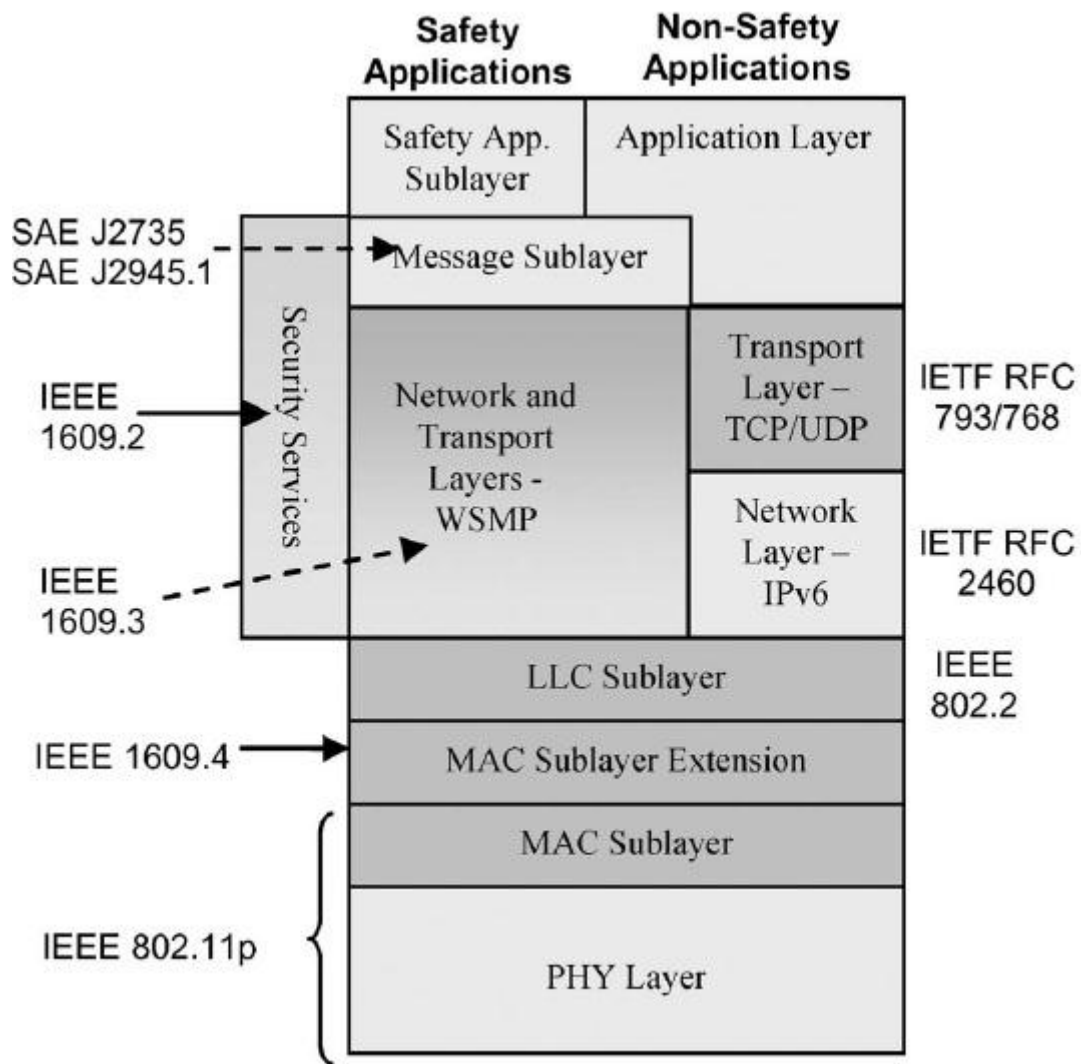


Figure 16: DSRC layered architecture [13]

DSRC physical layer

The DSRC physical layer uses IEEE802.11p which is an extension of IEEE802.11. To multiplex data the physical layer uses Orthogonal Frequency Division Multiplexing (OFDM) technology. The reasons of using OFDM are its high efficiency, good performance on multi-path routing and simple transceiver design. OFDM has three possible channels: 5MHz, 10MHz and 20MHz. DSRC generally uses the 10MHz channel while most IEEE802.11 implementations use the 20MHz channel. DSRC has a total of seven channels with six service channels and one control channel. The control channel handles safety related messages while the service channels takes care of the non-safety related messages [26]. Figure 17 shows the DSRC band plan.

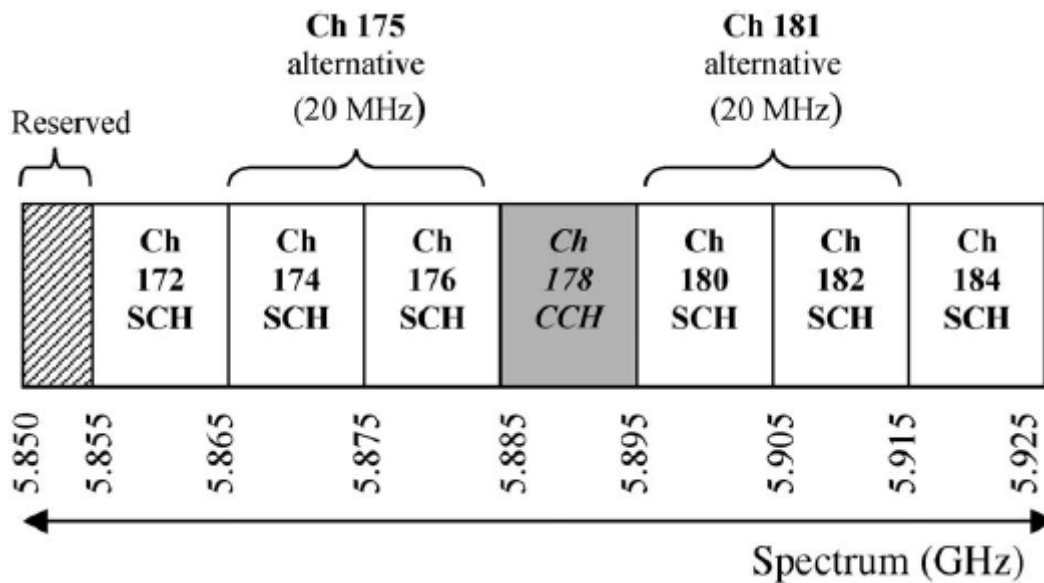


Figure 17: DSRC band plan [14]

IEEE1609.4 for multi-channel operation

This is an extension of MAC and applicable when DSRC operates in multi-channel mode. The target of the IEEE1609.4 standard is to define a mechanism to find the devices which are switching among multiple channels at the same time. The solution has two concepts: control channel (CCH) and service channel (SCH). As shown in above Figure 17, only one channel uses CCH treated as special channel. The device tunes the CCH in a regular interval. All other channels are SCH. The following Figure 18 shows the time division of CCH and SCH in IEEE1609.4. The synchronization time for each period is by default 100 ms. Each sync period is a combination of CCH followed by a SCH. By default the time division for CCH and SCH is 50 ms for each sync period. Each CCH and SCH starts with a 4 ms guard interval to allow the switching device to transfer control from one virtual MAC to another virtual MAC. Switching to the SCH is performed if the device is notified via a WSA (WAVE Service Advertisement). Generally switching is performed at the end of the CCH interval and returns to the next CCH interval. IEEE1609.4 also provides a so-called 'immediate departure' option which allows

switching to the SCH immediately after receiving WSA. It provides an ‘extended departure’ option as well which allows to remain in SCH for more sync periods until the service is completed.

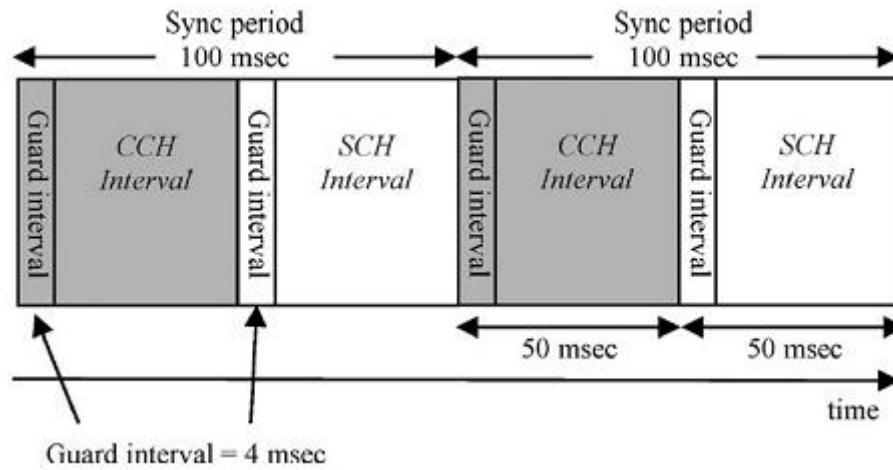


Figure 18: Time division between CCH and SCH interval [25]

IEEE1609.3 for network services

The IP (Internet Protocol) is a well-known protocol used in the network layer in many networks, especially the nodes which are connected to other nodes as a public internet. The benefit is to find a path to a node anywhere through its public IP address. However, for vehicular environment as packets are sent directly via airframe from source to destination, routing is not an issue. This also leads to less overhead of the packet. For this purpose the IEEE working group introduced WSMP (WAVE Short Message Protocol) in layer 3. WSMP sends WSMs (WAVE Short Messages). Figure 19 illustrates the WSM format.

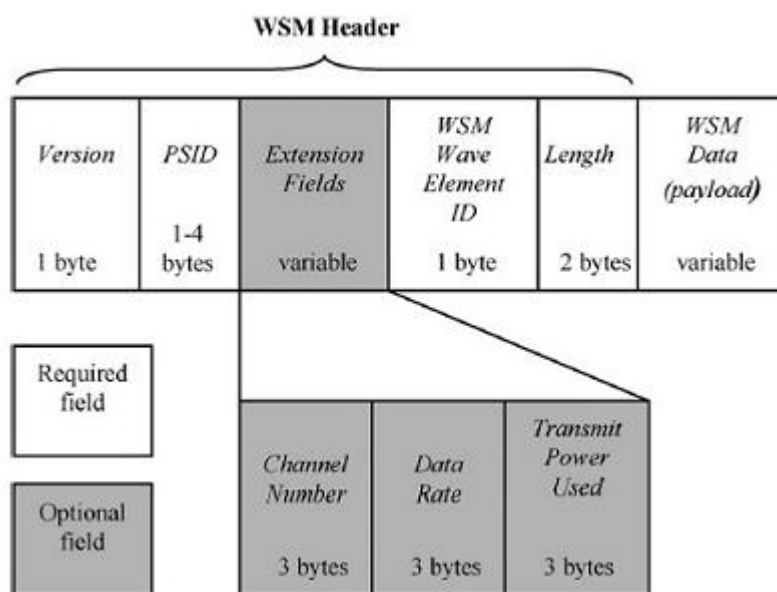


Figure 19: WAVE Short Message format [15]

Version

This is a 1 byte mandatory field and contains 4 bits WSMP version number and 4 reserved bits. In the current IEEE1603.3 standard the version number is 2. If the version number is higher than the designed version, the receiver will discard the WSM message.

Provider Service Identifier (PSID)

This is a unique mandatory field of 1-4 bytes. The device creates a list of allowed PSIDs for different services. When a WSM arrives, the application matches the received WSM PSID with the list of PSIDs. If a match is found, then a WSM packet is sent for the processing otherwise the application discards the WSM packet.

Extension Fields

It is a variable length optional field consisting of three fields: three bytes Channel Number which identifies the used channel (Figure 17), three bytes Data Rate and three bytes Transmit Power.

WSM WAVE Element ID

WSM WAVE element ID is a mandatory 1 byte field. The value of this field identifies the type of element in the WSM header. (See Appendix C for available elements and their corresponding values).

Length

The mandatory two bytes are the final bytes of the WSM header. The lower four bits define the length of the WSM data field in bytes. The upper four bits are reserved.

WSM Data

This is the payload of the WSM packet. For senders in general, WSM data are provided by the upper layers and in case of the receiver's data are passed to upper layers for processing.

IEEE1609.2 for middle layer security services

IEEE1609.2 standard defines the security services for application and management messages. It mainly focuses on the authentication of vehicle safety related messages (BSM: Basic Safety Message) in the WSM packet.

The development of the thesis is done in the OMNET++ simulator using Veins and SUMO simulation models. OMNET++ 4.4.1, Veins 2.2 and SUMO 0.17.1 are used for the implementation. The following two sections describe these two simulation models.

4.3. *Veins (Vehicles in Network Simulation) Simulation model*

Veins is an open source framework for Inter-Vehicular Communication (IVC). The simulation framework is combined of two simulators: OMNET++ for network simulation along with MiXiM model which runs in the physical layer and SUMO for road traffic simulation. The main benefit of the Veins simulation framework is, it supports bidirectional coupling between network simulation and road traffic microsimulation. Figure 20 depicts the Veins simulation framework which works on top of the MiXiM simulation model.

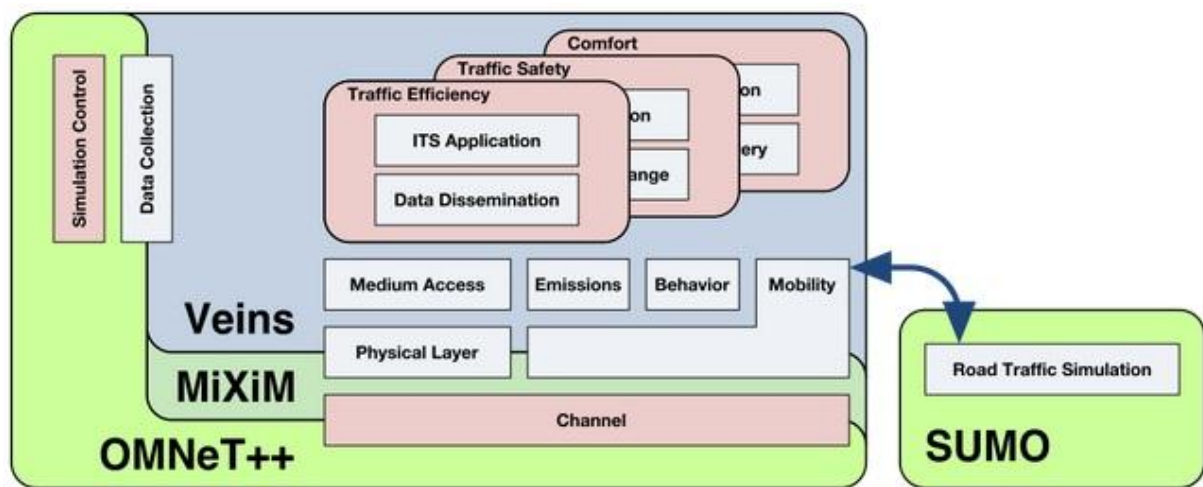


Figure 20: Veins simulation framework

Both simulators are connected via TCP socket and running in parallel. The communication between these two simulators is done via Traffic Control Interface (TraCI).

Network Simulation

Network simulation is used to model networks (computer networks or newly developed network protocols) and evaluate these before deployment in the real world. The benefit of network simulation is, performances of networks are evaluated before making the expensive field test evaluation. OMNET++ provides a list of reusable modules written in C++. The relationships of different modules and their communication is written in a network description file (NED). The NED file can be modeled graphically too. Simulation can be executed either from graphical environment or from command-line interface. Mobility patterns supported by OMNET++ are very simple and cannot be used for the VANET scenarios. The node mobility patterns are either obtained by the observation of real traffic or by using microsimulation model described in the next section.

Road Traffic Microsimulation

To achieve the realistic simulation of the moving nodes, the mobility should be modelled according to trace files obtained in real-world measurements. If the trace files are created using this approach, even then these trace files can only be supported for a specific scenario. Hence, it is necessary to generate movement traces files by simulation tools on the fly. Accordingly, the examined traffic flow road traffic simulation is classified into three parts: Macroscopic, Mesoscopic and Microscopic. VANET scenarios require accurate modelling of single radio transmission between nodes which implies that accurate node positions are needed. Both Microscopic and Mesoscopic simulation do not guarantee this feature. Only the Microscopic model which models the behavior of individual nodes and interaction between them is considered as the mobility model for VANET node simulation. The SUMO simulation model is used for the simulation of road and traffic environment. SUMO provides high performance simulation of huge road network including multiple lanes, junctions, and traffic lights. The vehicles in different roads and lanes are also configurable in SUMO. The section 4.4 describes in details about the features of SUMO simulation model.

Bidirectionally Coupled Simulation

By extending dedicated communication for both modules OMNET++ and SUMO, bidirectional coupling is achieved. During simulation, both models exchange commands and traces via TCP connection. According to Figure 21, the integrated control module is able to buffer any commands coming in between time steps to guarantee synchronous execution. At each timestep OMNET++ sends all buffered commands to SUMO and triggers the

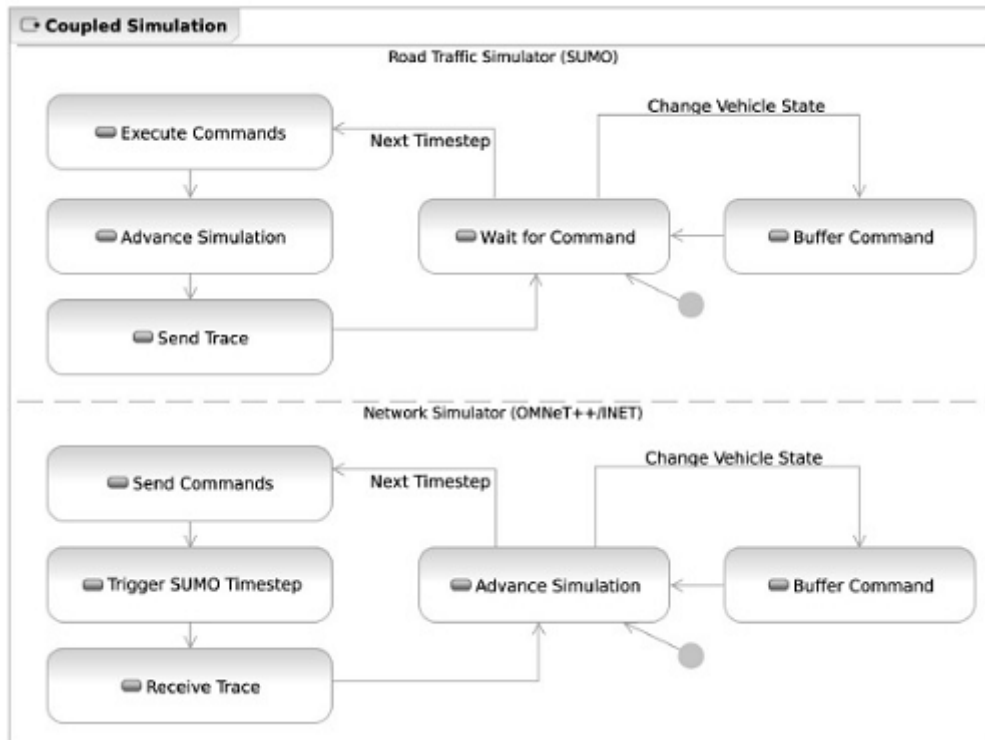


Figure 21: State machine of the OMNET++ and SUMO communication modules [24]

corresponding timestep of the SUMO. After completion, SUMO sends back commands and the position of the vehicles to OMNET++. OMNET++ reacts to the received mobility information of new vehicles, deleted vehicles which have been reached their destination and the new position of the existing vehicles. After processing the mobility information, OMNET++ advances the simulation until the next timestep so that vehicles can react to the modified environmental condition.

Figure 22 shows the message exchange between OMNET++ and SUMO in a sequence diagram. As described earlier, OMNET++ sends all commands to SUMO and then triggers the SUMO timestep. After execution SUMO sends back the trace data to OMNET++ which advances the simulation.

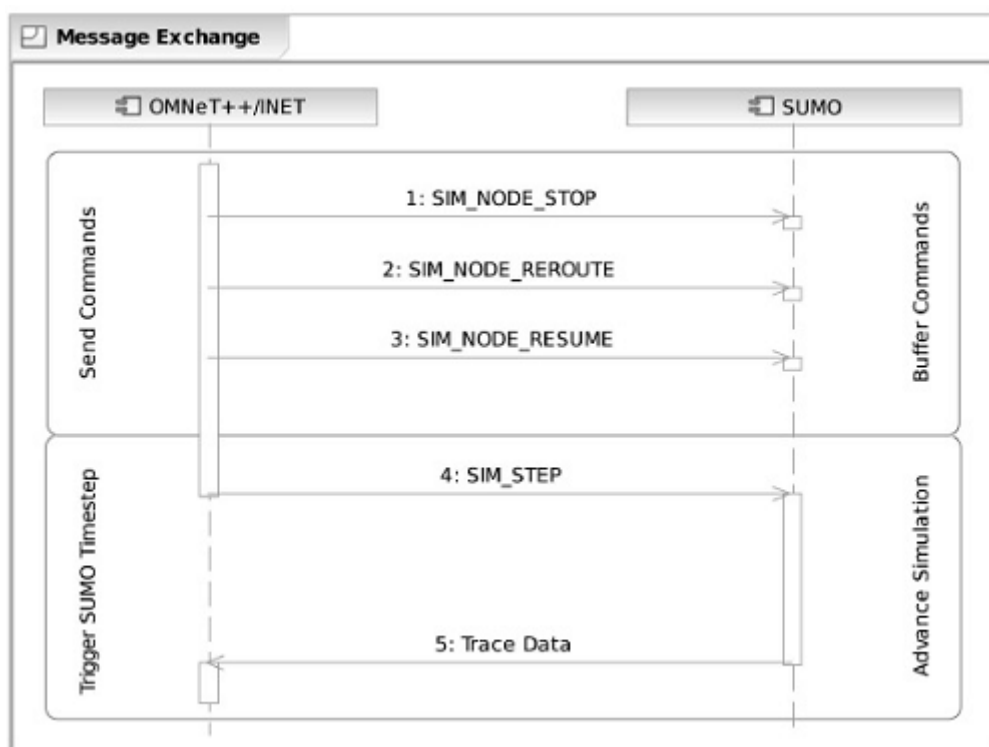


Figure 22: Sequence diagram of message exchange between OMNET++ and SUMO [24]

4.4. SUMO (Simulation of Urban MObility)

Simulation of Urban Mobility is an open source, microscopic simulation package to handle large road networks. It allows to simulate how a single vehicle moves through the defined road network. It includes several applications to prepare and simulate road and traffic environments. The simulation includes time-discreet vehicle movement, several vehicle types, multi-lane roads, traffic lights, etc. It can manage road networks with several 100,000 edges (combination of edges make a road) as well as fast execution (up to 100,000 vehicles updates per second on a 1GHz machine). It allows to import several types of network (VISUM, Vissim, Shapefiles, OSM, RoboCup, MATsim, openDRIVE and XML-Descriptions). The following table shows the different applications used in the SUMO simulation package to prepare the simulation.

Application	Purpose
<i>SUMO</i>	The command-line application for microscopic simulation
<i>SUMO-GUI</i>	Graphical user interface for the simulation
<i>NETCONVERT</i>	Imports network from different formats mentioned above and converts them to SUMO format
<i>NETGENERATE</i>	Generate road networks from the defined files (e.g. node, edge file)
<i>DUAROUTER</i>	Imports different description and computes the fastest routes of the network
<i>JTRROUTER</i>	Computes route from junction turning percentage
<i>DFROUTER</i>	Computes route from induction loop measurement
<i>OD2TRIPS</i>	Converts Origin-Destination matrices to the list of vehicle trips
<i>POLYCONVERT</i>	Imports polygons and point of interest from the different formats and converts them into SUMO visualized format
<i>ACTIVITYGEN</i>	It takes the definition of road-networks generated by NETCONVERT or NETGENERATE and generates vehicle types and route definitions

There are three mandatory files required for a SUMO simulation (configuration file, network file and routing file). The configuration file holds the path of the network and routing files. At the time of simulation SUMO only needs to know the path of the configuration file. The network file is generated by NETCONVERT from the node and edge file. Figure 23 shows the steps required to simulate road and traffic in SUMO.

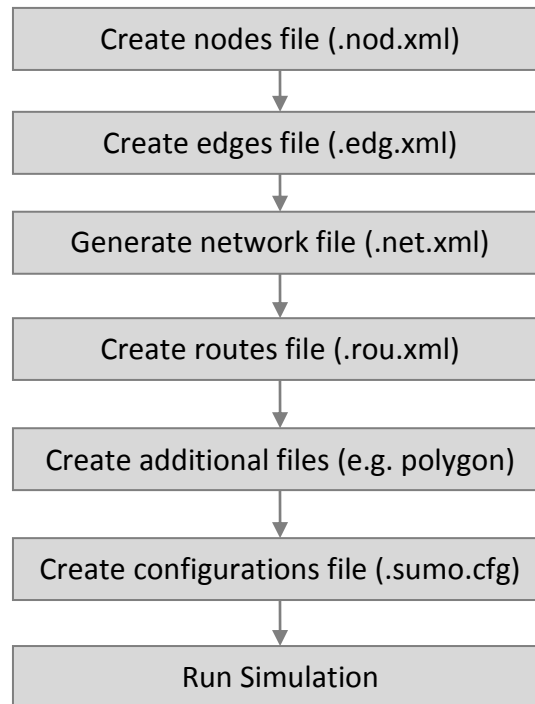


Figure 23: SUMO simulation process

The sample code snippet of mandatory xml files (node, edge, route and configuration file) are shown below. Node and edge files are required to generate network file (*.net.xml). As network file can be generated through “NETCONVERT”, sample code snippet for the network file is not provided here.

Nodes file (hello.nod.xml)

```
<nodes>
  <node id="1" x="-500.0" y="0.0" />
  <node id="2" x="+500.0" y="0.0" />
  <node id="3" x="+501.0" y="0.0" />
</nodes>
```

Edges file (hello.edg.xml)

```
<edges>
  <edge from="1" id="1to2" to="2" />
  <edge from="2" id="out" to="3" />
</edges>
```

Routes file (hello.rou.xml)

```
<routes>
  <vType accel="1.0" decel="5.0" id="Car" length="2.0" maxSpeed="100.0" sigma="0.0" />
  <route id="route0" edges="1to2 out"/>
  <vehicle depart="1" id="veh0" route="route0" type="Car" />
</routes>
```

Configurations file (hello.sumo.cfg)

```
<configuration>
  <input>
    <net-file value="hello.net.xml"/>
    <route-files value="hello.rou.xml"/>
  </input>
  <time>
    <begin value="0"/>
    <end value="10000"/>
  </time>
</configuration>
```

4.5. Development of Layered architecture

Figure 24 illustrates the implemented layered architecture using DSRC/WAVE on the upper layer and IEEE802.11p on the physical layer. DSRC uses MAC 1609.4 on the MAC layer.

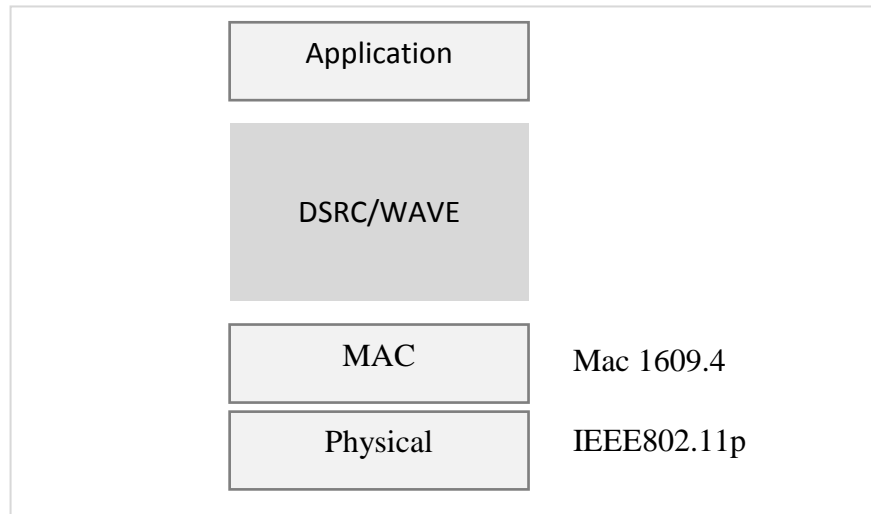


Figure 24: Layered architecture implementation

To implement the layered architecture the Veins simulation model is used (OMNET++ simulator). The Veins simulation model supports IEEE802.11p on the physical layer and DSRC/WAVE on the upper layer. Both vehicle and roadside unit are using the same layered architecture. The roadside unit creates a wave short message (WSM) packet containing the speed limit as data and broadcasts it to the network. On the vehicle side, when it receives the broadcast message, it extracts the speed limit from the WSM packet and applies the received speed to the vehicle. The roadside unit sends speed limit for both directions.

Some key notes before describing the implementation in details

Network description file: Necessary file for the OMNET++ simulator to define the network. Parameters and modules are defined in the network description file. A module can also be a network description file type.

omnetpp.ini file: Configuration file for the project. It contains all the required information (e.g. carrier frequency, packet size, data rate, power consumption). The OMNET++ project is built based on the parameters and features defined in this file. Parameters and modules settings defined in the network file can be manipulated here. Several networks can be handled here.

Traci Scenario Manager: A module that takes care of the communication between OMNET++ module and SUMO generated road and traffic network.

Step by step process in short:

1. An OMNET++ project is created using Veins simulation model. Figure 28 shows the excerpt of the file structure of the created OMNET++ project. Rather than showing the empty project, the figure illustrates the final file structure after implementation.
2. As IEEE802.11p is used in the physical layer and DSRC/WAVE on the upper layer; carrier frequency 5.89 GHz, bit rate 18 Mbps and header length for message is 256 bits defined. The following Figure 25 shows the part of the omnetpp.ini file which includes the definition of IEEE802.11p and MAC 1609.4 parameters.

```
#####  
#               11p specific parameters               #  
#                                                     #  
#               NIC-Settings                           #  
#####  
*.connectionManager.pMax = 20mW  
*.connectionManager.sat = -89dBm  
*.connectionManager.alpha = 2.0  
*.connectionManager.carrierFrequency = 5.890e9 Hz  
*.connectionManager.sendDirect = true  
  
*.*.nic.mac1609_4.useServiceChannel = false  
  
*.*.nic.mac1609_4.txPower = 20mW  
*.*.nic.mac1609_4.bitrate = 18Mbps  
  
*.*.nic.phy80211p.sensitivity = -89dBm  
*.*.nic.phy80211p.maxTXPower = 10mW  
*.*.nic.phy80211p.useThermalNoise = true  
*.*.nic.phy80211p.thermalNoise = -110dBm  
*.*.nic.phy80211p.decoder = xmldoc("config.xml")  
*.*.nic.phy80211p.analogueModels = xmldoc("config.xml")  
*.*.nic.phy80211p.usePropagationDelay = true
```

Figure 25: IEEE802.11p and MAC1609.4 parameters definition

3. The Veins simulation model supports IEEE802.11p on the physical layer, MAC 1609.4 on the MAC layer and DSRC/WAVE on the upper layer. All these layers are implemented in this project as well as necessary mechanisms for communication on the layers. WSM (wave short message) holds the data required to transmit to another module.
4. RSU and vehicle are defined as a modules in the different network description files. The functionalities (e.g. broadcasting speed limit by RSU, receiving broadcast messages by vehicles) of both modules is defined in different C++ files. Figure 26 and Figure 27 show the ned (network description) file of the car and rsu respectively.

```

module Car
{
    parameters:
        string applType; //type of the application layer
        string mobilityType; //type of the mobility module
    gates:
        input radioIn; // gate for sendDirect
    submodules:
        appl: <applType> like IBaseApplLayer {
            parameters:
                @display("p=60,50");
        }

        nic: Nic80211p {
            parameters:
                @display("p=60,166");
        }

        mobility: <mobilityType> like IMobility {
            parameters:
                @display("p=130,172;i=block/cogwheel");
        }
    connections:
        nic.upperLayerOut --> appl.lowerLayerIn;
        nic.upperLayerIn <-- appl.lowerLayerOut;
        nic.upperControlOut --> appl.lowerControlIn;
        nic.upperControlIn <-- appl.lowerControlOut;

        radioIn --> nic.radioIn;
}

```

Figure 26: Car ned file

```

module RSU
{
    parameters:
        string applType; //type of the application layer
        int vehicleSpeed;
        int revVehicleSpeed;

        int r0VehicleCounter;
        int r0RevVehicleCounter;
        int r1VehicleCounter;
        int r1RevVehicleCounter;
    gates:
        input rsuradioIn@directIn; // gate for sendDirect
    submodules:
        appl: <applType> like IBaseApplLayer {
            parameters:
                @display("p=60,50");
        }

        nic: Nic80211p {
            parameters:
                @display("p=60,166");
        }

        mobility: BaseMobility {
            parameters:
                @display("p=130,172;i=block/cogwheel");
        }
    connections:
        nic.upperLayerOut --> appl.lowerLayerIn;
        nic.upperLayerIn <-- appl.lowerLayerOut;
}

```

Figure 27: RSU ned file

5. After defining the network description files, it is required to write C++ code for both car and RSU to handle the situation. RSU C++ file set the speed limit in WSM packet as WSM data. Car C++ file extracts the speed limit from the WSM data. The RSU sets speed limit for both directions at the same time as one WSM data. After extracting the speed limit from WSM data, the vehicle applies the corresponding speed limit. If the vehicle is in normal direction, it will apply the speed limit for the normal direction and vice-versa. Code snippets for both car and RSU are shown below:

RSU.cc

```
mRsu->par("vehicleSpeed") = vehicleSpeed;
mRsu->par("revVehicleSpeed") = revVehicleSpeed;
vs1 = mRsu->par("vehicleSpeed");
rvs1 = mRsu->par("revVehicleSpeed");
std::stringstream stream;
stream << vehicleSpeed << " " << revVehicleSpeed;
std::string strVehicleSpeed = stream.str();
wsm->setWsmData(strVehicleSpeed.c_str());
sendWSM(wsm);
```

Car.cc

```
std::stringstream stream;
stream << std::hex << speedLimit;
std::string hex_val = stream.str();
int32_t value = atoi(hex_val.c_str());
traci->commandSetSpeedMode(value);
traci->commandSetSpeed((double)speedLimit);
```

6. To maintain the speed limit for different scenarios, the tables in section 4.5 are used corresponding to the traffic congestion. The initial speed limit is set at the start of the simulation. The speed limit is updated later considering the above mentioned table and RSU starts to broadcast message every 3 seconds. When a RSU broadcast as new speed limit, all vehicles in range receive the speed limit and react on it. Based on the number of vehicles in range, the RSU sets a new speed limit and broadcasts it to the vehicles.
7. The implementation is done using one general scenario, which has a straight line road with vehicles in both directions. After initial implementation, two more scenarios are integrated. Two RSUs are set for each scenario. In the next chapter scenarios will be described in details.

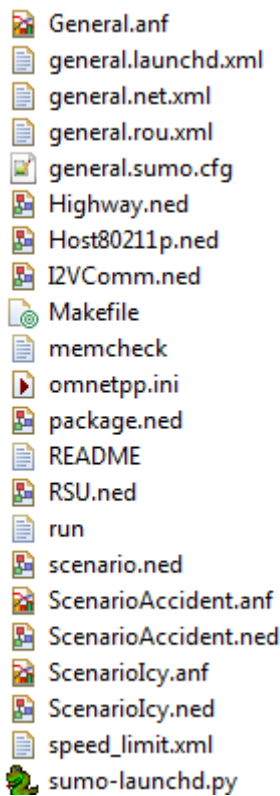
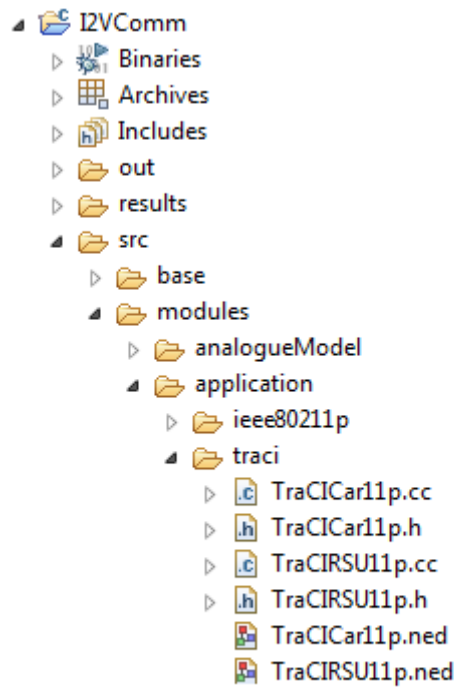


Figure 28: File structure of the omnet++ project

Figure 29 illustrates the communication in a sequence diagram. At the beginning the Roadside Unit transmits the legal speed limit. Then, based on the scenario and traffic congestion the RSU calculates the new speed limit and broadcasts it to approaching vehicles. This is a continuous process which runs every 3 seconds. For the simulation purpose, it runs on every 3 seconds so that when the simulation starts communication between roadside unit and vehicles are well understandable.

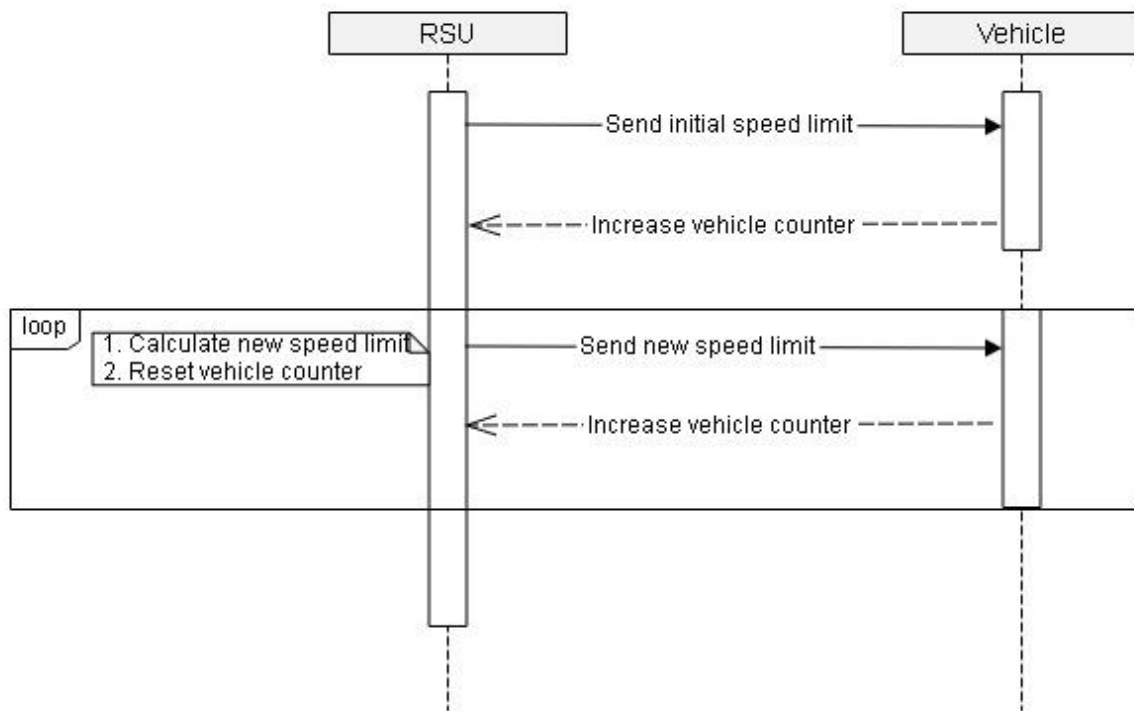


Figure 29: Sequence diagram of communication

4.6. Road network and traffic flow

There is a total of three scenarios to simulate the developed layered architecture. Three scenarios are used to check that the implementation meets its requirements on different roads and environmental conditions (e.g. icy road, accident ahead). The three scenarios are general, icy and accident scenario. All the scenarios have a road network of single lane and traffic in both directions. The general scenario constructs a straight line road while icy and accident scenarios have a curvy road. The icy scenario has curves in top direction while the accident scenario has curves in bottom direction. In the general scenario vehicles receive the legal speed limit as a beacon from roadside unit. The icy scenario assumes for a certain area the road surface is icy for both directions between roadside unit 1 and roadside unit 2. In case of icy scenario RSU0 advises a smaller speed limit for the reverse direction vehicle while RSU1 sends the same speed limit to the normal direction vehicles. In case of accident scenario, it assumes that there is an accident in the reverse direction which causes partial

blocking of the road. For that reason, RSU0 advises a new speed limit for the vehicles of the opposite direction. The used road networks are shown in the following.

Road network for general scenario:

Figure 30 shows the road network for the general scenario. The network has a straight road with both directions. Each direction has a single lane. There are two types of vehicles running in the road. In each direction a total of 90 vehicles of the same type are running. The maximum speed is set to 100km/h.



Figure 30: Part of the general road scenario

Road network for icy scenario:

Figure 31 shows the road network used by the icy scenario. This is a curvy road in direction to top. The road network on the above diagram is used for the icy road condition. The other parameters like vehicle flow and speed, number of vehicles are the same as with the general road network.

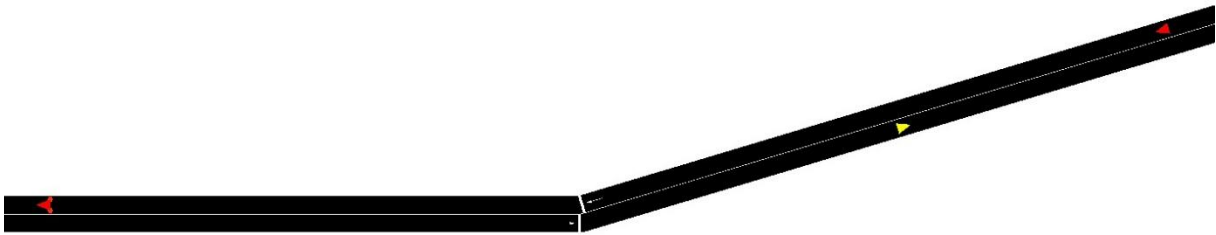


Figure 31: Part of icy road scenario

Road network for accident scenario:

Figure 32 shows an accident scenario road network. This road is also a curvy road like in the icy scenario, but the curve is going in bottom direction. The other parameters like vehicle flow and speed, number of vehicles are the same as with the general road network.

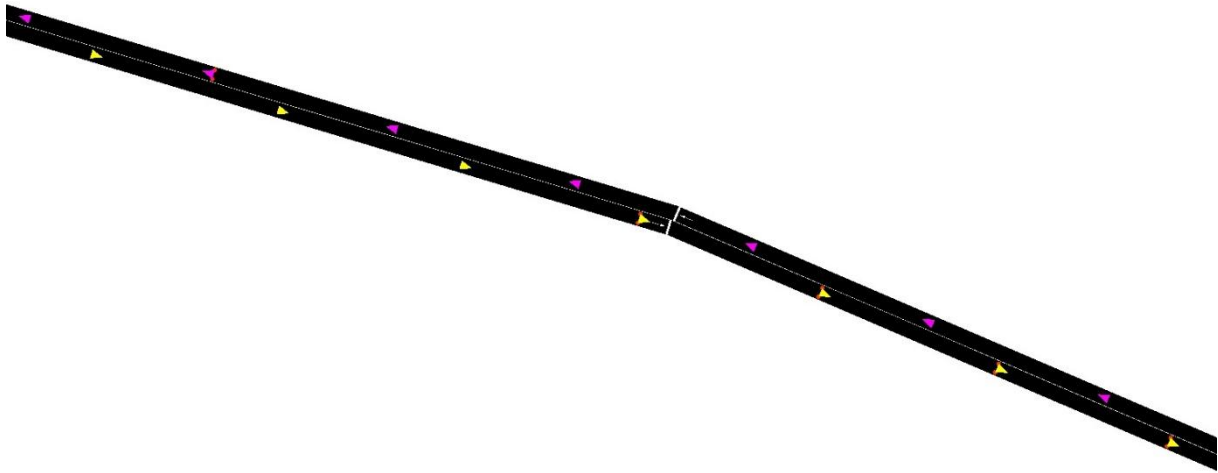


Figure 32: Part of accident road scenario

4.7. Scenario integration in OMNET++

The road networks mentioned in section 4.6 are integrated as different scenarios in the main OMNET++ project. Scenarios are configured in the ini file. When the simulation starts, a dialog box appears and asks the user to choose a scenario (please find in Appendix B).

In the general scenario, the maximum lane speed is set to 100km/h as well as the maximum vehicle speed. When RSU broadcasts the initial speed limit, the first RSU broadcasts a speed limit of 60km/h for both directions and the second RSU broadcasts a speed limit of 90km/h for both directions. The initial speed limit is set to ini file. Also the network description xml files are set in the configuration file as a path for Traci Scenario Manager.

For the icy scenario, the maximum lane speed and the maximum vehicle speed are set as general scenario. But, the vehicle speed is set to 40km/h by the first RSU and 90km/h by the second RSU.

As well as for accident scenario, the maximum lane speed and the maximum vehicle speed is the same as with general scenario. Assuming there is an accident on the reverse direction, initial vehicle speed is set to 90km/h (normal direction), 40km/h (reverse direction) by first RSU. The second RSU sets the vehicle speed limit to 90km/h taking into the consideration that vehicles crossed the accident area.

After detecting the number of vehicles in the range the new speed limit is set by the RSU considering the following tables. The tables contain the speed limit based on the number of vehicles in the range of the RSU.

Speed limit for general scenario:

Density (no. of cars in transmission range)	Normal direction speed limit (km/h)		Reverse direction speed limit (km/h)	
	RSU0	RSU1	RSU0	RSU1
<= 2	100	100	100	100
<= 4	80	80	80	80
<= 6	60	60	60	60
<= 8	50	50	50	50
<= 10	50	50	50	50
> 10	40	40	40	40

Speed limit for icy scenario:

Density (no. of cars in transmission range)	Normal direction speed limit (km/h)		Reverse direction speed limit (km/h)	
	RSU0	RSU1	RSU0	RSU1
<= 2	100	60	60	100
<= 4	80	55	55	80
<= 6	60	50	50	60
<= 8	50	45	45	50
<= 10	50	45	45	50
> 10	40	40	40	40

Speed limit for accident scenario:

Density (no. of cars in transmission range)	Normal direction speed limit (km/h)		Reverse direction speed limit (km/h)	
	RSU0	RSU1	RSU0	RSU1
<= 2	100	100	55	100
<= 4	80	80	55	80
<= 6	60	60	50	60
<= 8	50	50	45	50
<= 10	50	50	40	50
> 10	40	40	35	40

5. Simulation

After implementation and integration of scenarios, it is necessary to simulate the scenarios on OMNET++. For each scenario, there are two RSUs. The first RSU is placed on the top right and other RSU is placed on the bottom left. Each RSU is broadcasting its speed limit on every 3 seconds. Each RSU transmits messages to vehicles of both directions. When the vehicles receive the speed limit, they react immediately. Vehicle flow and direction is defined in the SUMO routing files. The simulations of different scenarios are shown below.

5.1. General scenario simulation

Figure 33 shows the OMNET++ simulation of the general scenario. As described in section 4.6 the scenario has a straight road. So, the vehicles are driving straight in both directions. The blue dotted line shows the broadcasting of airframe packets from RSU0 to the vehicles in its range. In both directions both RSU0 and RSU1 broadcast the legal speed limit. The vehicles which have already received the initial speed limit are marked as green.

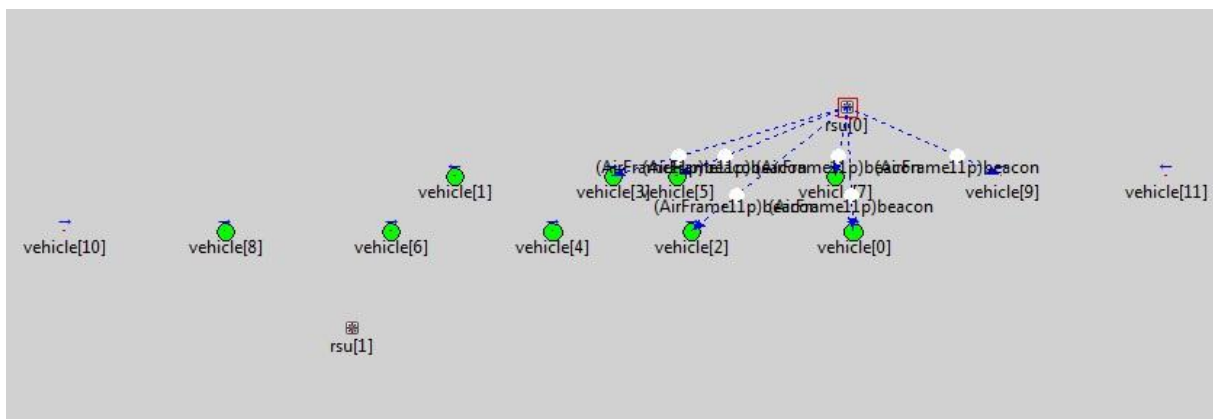


Figure 33: General simulation

5.2. Icy scenario simulation

Figure 34 shows the simulation of the icy scenario. As the scenario has a curve road on the upper direction, the vehicles are driving in the defined directions. At a certain time RSU1 broadcasts the message to the vehicles which is shown with the blue dotted line. As described in section 4.6, the icy road area lies between RSU0 and RSU1. For this, RSU1 broadcast the new calculated speed limit for the normal direction vehicle. RSU0 does the same for reverse direction vehicles. Both RSU0 and RSU1 broadcast the legal speed limit based on the traffic congestion for the other direction.

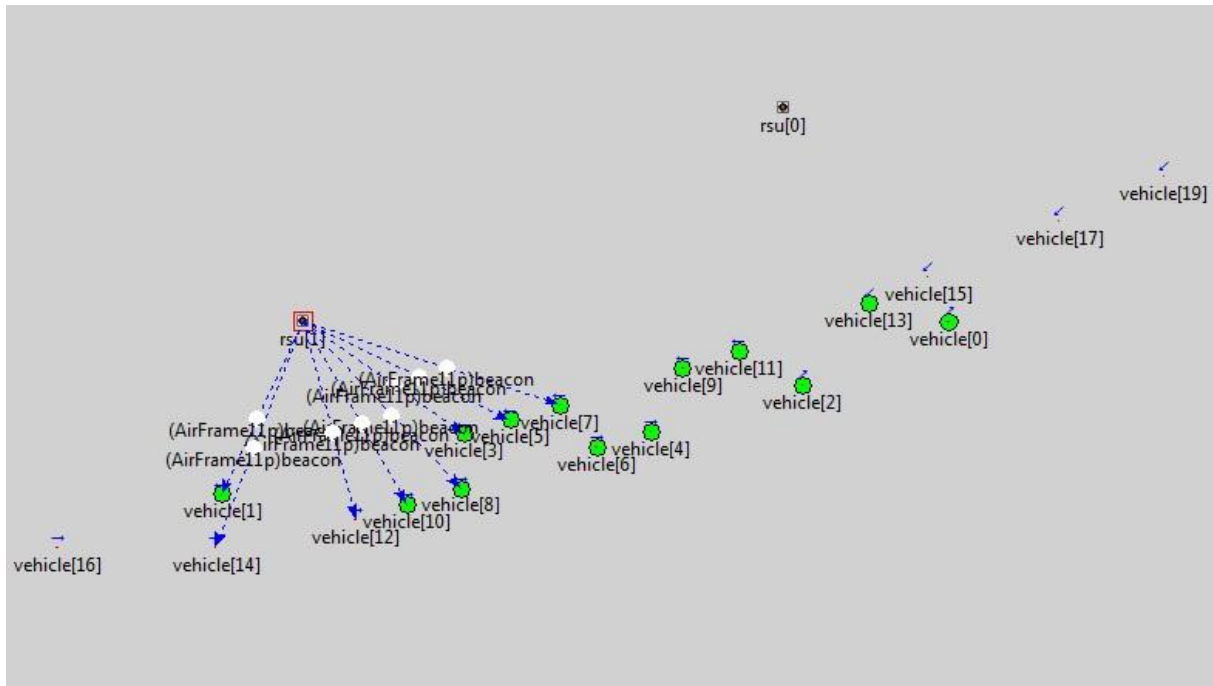


Figure 34: Icy simulation

5.3. Accident scenario simulation

Like with the previous two simulations, Figure 35 shows the simulation behavior of the accident scenario. The indication of an accident in the road is shown by a symbol in the middle of the road. RSU0 broadcasts the speed limit by considering the accident ahead. As the accident area is in the reverse direction, only RSU0 broadcast the new calculated speed limit for the reverse direction.

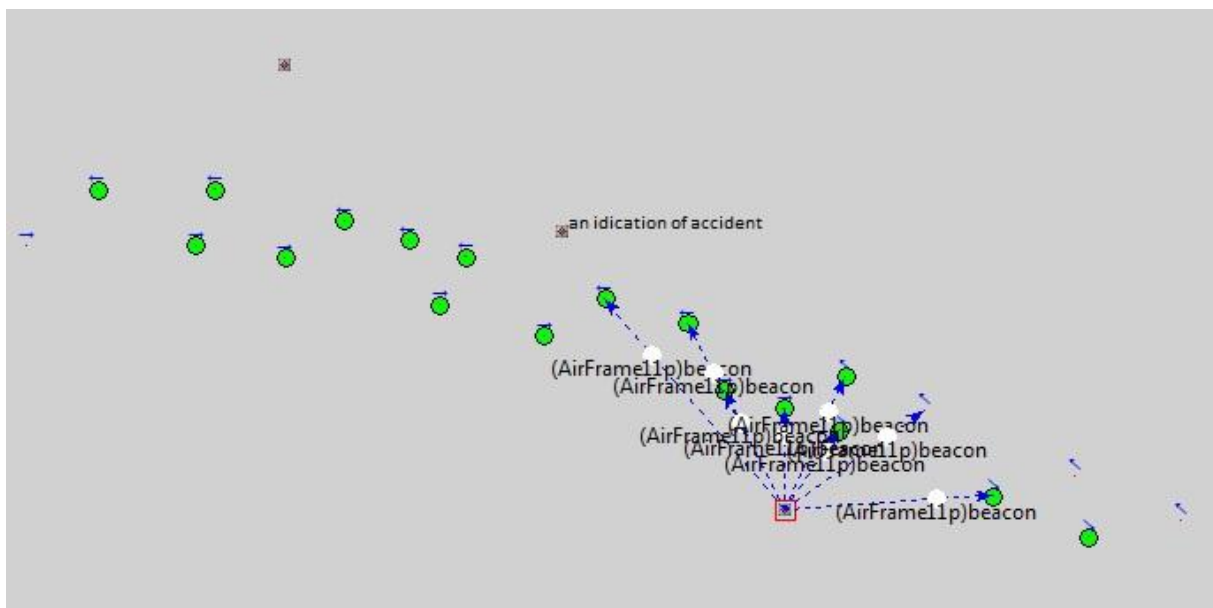


Figure 35: Accident simulation

5.4. Expected results

The expected results of the simulation can be divided into several parts as follows:

- The main expectation is that the vehicle speed should be decreased if the number of vehicles increases in the roadside unit range. That means the vehicles should drive slowly. Vice versa, if vehicle density decreases, then the vehicle speed should be increased and the vehicles should drive faster.
- The advised speed limit should match with the defined speed limit in section 4.5 for different scenarios.
- After crossing the special area of the road (e.g., icy road surface, accident area) the corresponding vehicle should get the normal speed limit based on the traffic congestion only.
- As a beacon is sent every 3 seconds, the vehicle counter (used for traffic congestion) should be reset every 3 seconds.
- The distance between vehicles should not be uniform.

6. Evaluation

All the results are logged in a file after successful simulation. Generated result files (graph view) are described for each scenario in details. From each scenario only one vehicle is taken into account for the result analysis. The results are logged in both vector and scalar file. Vector result file describes the result in details considering the time frame. But, the scalar result file contains generic information (e.g. total distance covered by a vehicle, maximum speed in the whole journey, etc.). The result files are generated for each vehicle.

6.1. Evaluation of general scenario

Figure 36 illustrates the behavior of a vehicle (vector analysis) on the road network in a graph.

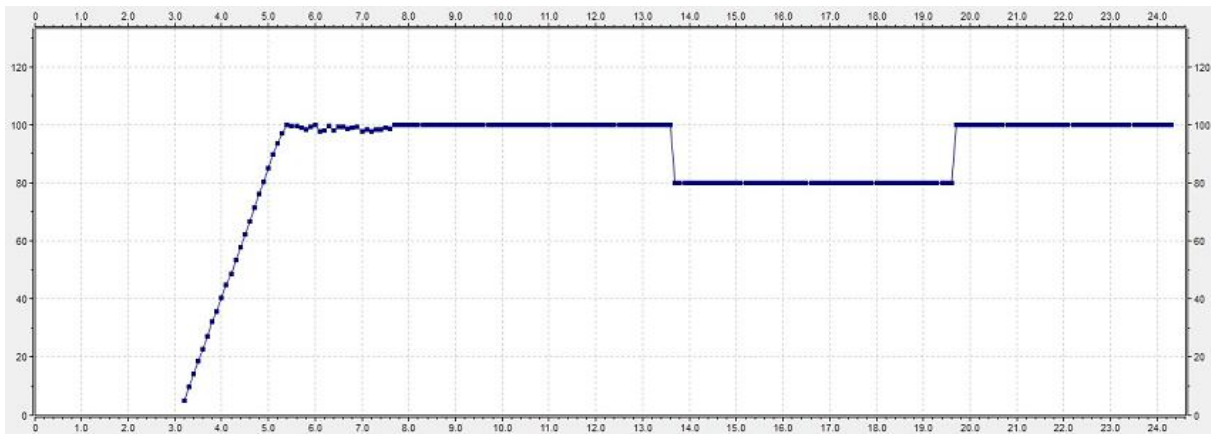


Figure 36: General scenario result evaluation

As the maximum speed limit of vehicle and lane is set to 100km/h, the vehicle goes up to 100km/h before receiving any messages from the road side unit. Around 8 seconds the vehicle gets the legal speed limit of 100km/h from the RSU1 (considering density-speed limit table in section 4.5) which is the same as the vehicle speed. Until 13.5 seconds the vehicle receives the same speed limit of 100km/h. At 13.5 seconds the vehicle receives a new speed limit of 80km/h from RSU1 based on the traffic congestion and applies the speed to the vehicle which starts driving slowly which is the same as the expected behavior that vehicle speed is inversely proportional to traffic congestion. Until 19.5 seconds it receives the same speed limit of 80km/h. At 19.5 seconds it receives the new speed limit of 100km/h and increases the vehicle speed to 100km/h. The vehicle got the new speed limit on every 3 seconds as well as advised speed limit matched with the congestion-speed table in section 4.5. The analysis indicates that simulation results conform to the expected results except the uniform distance between vehicles.

Figure 37 and Figure 38 show the scalar result of one vehicle of the general scenario. Here, only the speed information is shown. Besides the speed several other pieces of information

are available e.g. total time of travel, total distance covered, total packets sent and so on. Figure 37 shows the minimum speed and Figure 38 shows the maximum speed of vehicle1 in the whole journey.

The minimum speed of vehicle1 in the whole journey was ~ 4.82 km/h. As the minimum speed limit set by the application is 35 km/h, the logged minimum speed is the initial speed when the vehicle came into the simulation environment.

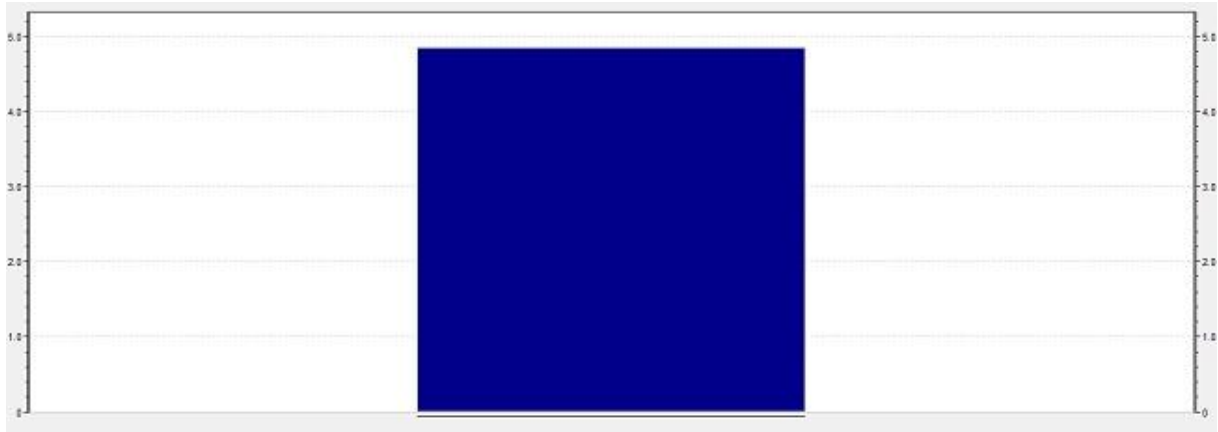


Figure 37: Minimum speed of vehicle1

The maximum speed limit of the road and vehicles is set to 100 km/h. The speed limit application also sets the maximum speed to 100km/h. For this reason, vehicle1 drives up to 100 km/h in the whole journey.

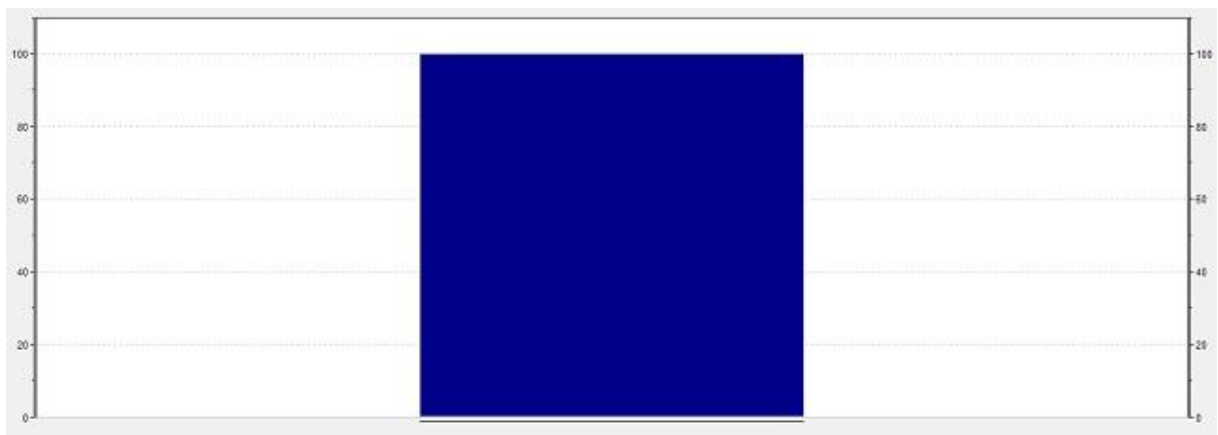


Figure 38: Maximum speed of vehicle1

6.2. Evaluation of icy scenario

It is assumed that the icy road is situated in between RSU0 and RSU1 in both directions. Figure 39 depicts the behavior of a vehicle (vector analysis) which has started its journey in the normal direction.

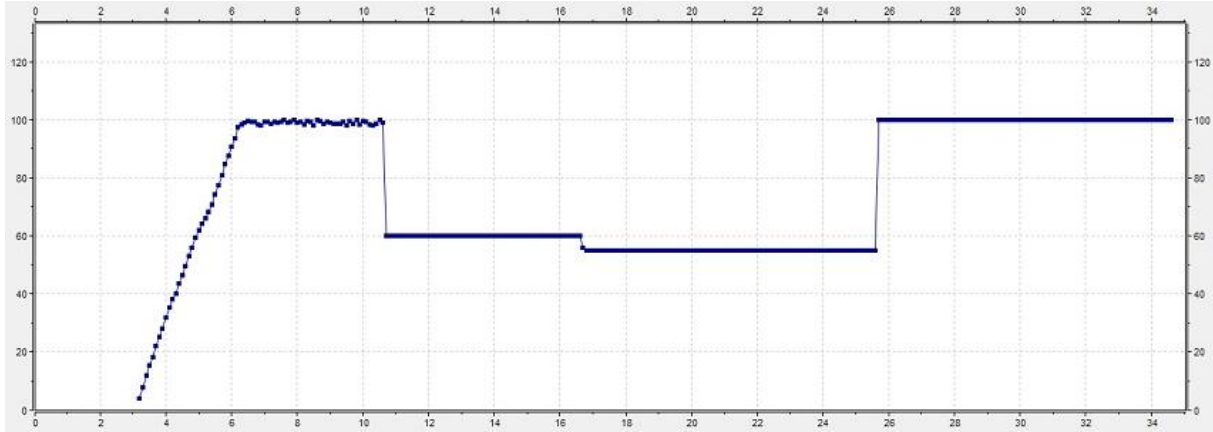


Figure 39: Icy scenario result evaluation normal direction

Like in the previous scenario, initially the vehicle goes to the max speed limit before receiving any speed limit message from the roadside unit. Around 10.5 seconds the vehicle receives an initial speed limit of 60km/h by RSU0, and slows down to the received speed limit. Around 16.5 seconds, based on the traffic congestion vehicle receives a different speed limit of 55km/h from the RSU0 and applies the received speed to the vehicle. As traffic congestion increases in the icy road area, it further slows down its speed to the advised speed, which implies that increase of traffic congestion results to drive slowly and hence satisfies the main expected result. Around 26 seconds the vehicle receives a new different speed limit of 100km/h from RSU1 and increases speed to 100km/h. At around 26 seconds the vehicle passed the icy road surface area and reached the range of RSU1, it receives the normal speed limit based on the congestion only. This result satisfies the expected behavior of the reception of new speed limit after passing the special road area.

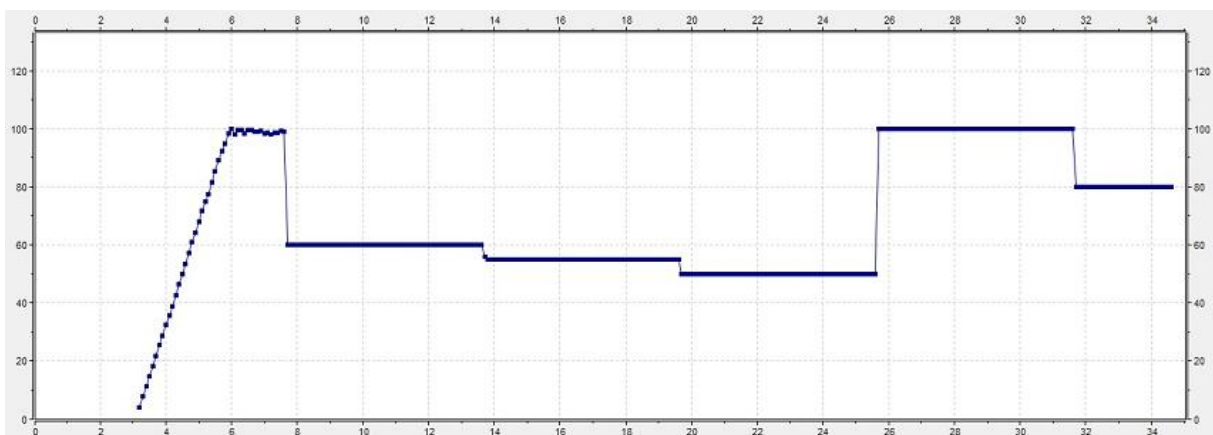


Figure 40: Icy scenario result evaluation reverse direction

Figure 40 shows the behavior of a vehicle in the reverse direction. Like as the other vehicles in the beginning it goes to max speed of 100km/h. Around 10 seconds the vehicle gets the new calculated speed limit of 60km/h for the first time according to the congestion-speed table mentioned in section 4.5. The application which runs on the vehicle decreases the speed to 60km/h. Around 14 seconds as the number of vehicles in the area increases compared to last time, the new calculated speed limit is set to 55km/h by RSU0. When the vehicle went further at the time around 20 seconds a new speed limit is set to 50km/h by RSU1 based on the traffic congestion. Accordingly, the vehicle speed is decreased to 55km/h which satisfies the expected behavior of vehicle speed with respect to traffic congestion. At the time, around 25 seconds the vehicle passed the icy road area and new speed limit is set by RSU1. As there is less traffic the speed is set to 100km/h. Around 32 seconds number of vehicles in the area increases and RSU1 sets the new speed limit to 80km/h. The approaching vehicle receives the new speed limit and starts driving slowly which implies that the result is the same as the expected behavior of getting normal speed when the vehicle passes the special area.

6.3. Evaluation of accident scenario

Figure 41 shows the behavior of a vehicle (vector analysis) which is running in the same direction where the accident occurred.

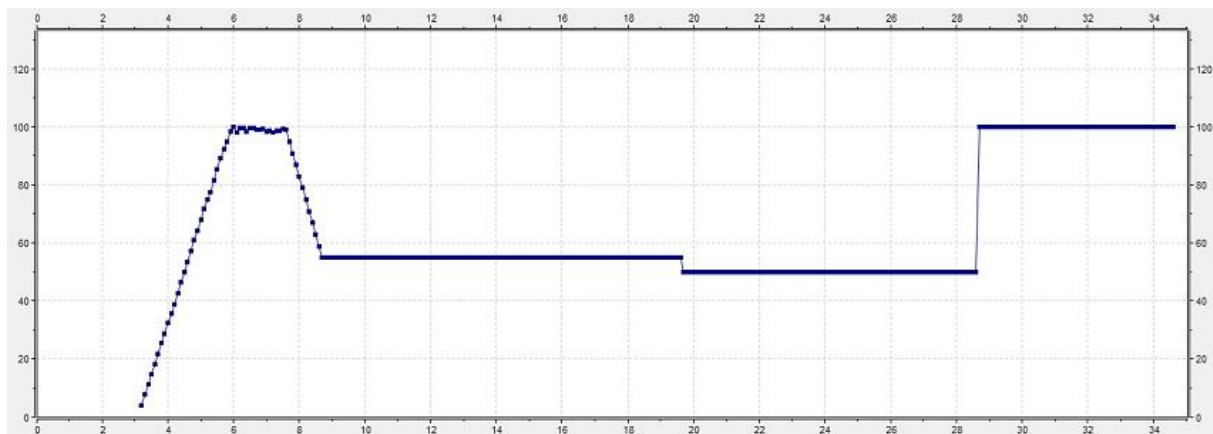


Figure 41: Accident scenario result evaluation

As mentioned before, it is assumed that the accident happens in the reverse direction of the road network between RSU0 and RSU1 which causes the partial blocking of the road. As with the other simulation scenarios at the beginning the vehicle goes to maximum speed of 100km/h. Around 8.5 seconds it gets the new speed limit of 55km/h because of the accident ahead which causes partial blocking of the road. At around 20 seconds the vehicle gets the new calculated speed limit of 50km/h based on the number of vehicles in the range of RSU0. The vehicle applies the new received speed limit and slows down to 50km/h. When a vehicle passes the accident area at around 29 seconds it receives a new speed limit of 100km/h

from RSU1 and increases the speed to 100km/h. The simulation results comply with the expected behavior.

The minimum speed for a vehicle in both icy and accident scenario is almost the same. Additionally, the maximum speed is also the same in these two scenarios. For this reason, the scalar analysis is not shown for icy and accident scenario.

6.4. Evaluation summary

The resulting analysis of the three different scenarios shows that the simulation results conform with the expected results except the uniform distance between the vehicles. The vehicles in each scenario slow down when the traffic congestion is increased. Approaching the incident area, vehicles drive slowly and after leaving the area vehicles drives faster. Traffic congestion has been taken into account in both cases. For most of the cases the distance between two vehicles is the same as the distance between two other vehicles.

6.5. Limitations of overall simulation

The simulation supports single lane road network for broadcasting the speed limits. It does not send different speed limits for the different lanes in a single direction. To cope with different speed limits for different lanes, it is necessary to implement the lane detection mechanism first.

The traffic congestion has been calculated based on the number of vehicles in the range of a roadside unit for a single direction. In real environment, traffic congestion is calculated by a device installed in the roadside which counts the number of vehicles passed in a certain amount of time.

The initial distance between two vehicles driving in the same direction is uniform. The distance remains the same between vehicles until the vehicles do not receive speed limit. The vehicles which have received the speed limit have a uniform distance from one to the next one.

The current simulation is tested only for a non-divided road network. This implies that the application behavior has not been tested for a road network where the road is divided by junctions.

7. Summary and Prospects

This thesis described different related projects focusing on roadside to vehicle communication. The high level architecture and communication mechanism for the thesis has been chosen based on the research of related projects. As the related projects are using IEEE802.11p in the physical layer and CALM technology in the upper layer, the architecture of the thesis has been designed according to the related projects. The main advantage of using CALM is, several communication media can be handled in the same application. Since CALM is still in development and clear specifications are not yet released, it was decided to use DSRC/WAVE in the upper layer for the implementation. In the transport and network layer DSRC uses WSMP (WAVE Short Message Protocol) for safety related applications and for non-safety applications DSRC uses TCP/UDP and IPv6 in the transport and network layer respectively. Since it has been decided to implement variable speed limit as an example application for the thesis which is a safety related application, the thesis uses DSRC/WAVE in the upper layer. Three scenarios (general, icy road and accident ahead) have been created and integrated to test the variable speed limit application. The speed limit for each scenario is based on the traffic congestion (predefined 'traffic congestion-speed' table). The simulation results are logged and analyzed to verify whether received results are in accordance with the expected results or not. The expected and simulation results matched, except the uniform distance between vehicles. For all three scenarios when the vehicles receive the new calculated speed limit, the application immediately slows down the car to the received speed. The maximum speed limit is set to 100 km/h. All the three scenarios received a new calculated speed limit every 3 seconds. For simulation the repetition of the broadcast by the roadside unit was set to 3 seconds, but is configurable in the OMNET++ configuration file.

For future applications there exist several possible improvements. The speed limit application may be changed in order to broadcast different speed limits for multiple lanes. But to achieve this goal, it is necessary to incorporate lane detection functionality in the application.

Traffic congestion information can be acquired from an installed roadside equipment. The equipment will transfer number of vehicles passed in a certain time interval.

Speed limit can be calculated by an introduced equation which will take the traffic congestion and calculates the speed limit and return result for broadcast.

The application can be extended to handle road networks with junctions. To implement this feature, may also be expedient to incorporate traffic lights feature into the application.

8. List of Abbreviations/Definitions

Term	Abbreviations
CALM	Communications, Air-interface, Long and Medium range
DAB	Digital Audio Broadcasting
DVB	Digital Video Broadcasting
HMI	Human Machine Interface
IVC	Inter-vehicular Communication
LDM	Local Dynamic Map
Ned	OMNET++ network description file
RSU	Road-Side Unit
TIC	Traffic Information Centre
TCC	Traffic Control Centre
V2I	Vehicle to Infrastructure
V2V	Vehicle to Vehicle
VANET	Vehicular Ad-hoc Network
Veins	Vehicles in Network Simulation
VMS	Variable Message Sign
WSM	WAVE Short Message
WSMP	WAVE Short Message Protocol

9. Table of Figures

Figure 1: Three models of traffic management system [1].....	3
Figure 2: Proposed interfaces between different devices in the COOPERS architecture [2]	4
Figure 3: System structure at Site-1 AT/IT [3]	5
Figure 4: CVIS high level architecture [4]	7
Figure 5: Hardware configuration of a test vehicle [5]	10
Figure 6: Architecture of the interconnections between vehicle, RSU, COMO Centre and Traffic management Centre (VZH) [6]	11
Figure 7: Speed limit [7]	12
Figure 8: VANET architecture (LDM and Message Generation) [8]	14
Figure 9: Infrastructure system components [9]	14
Figure 10: Three areas and ahead incident area (black) [10]	18
Figure 11: High Level Architecture	23
Figure 12: Layered Architecture concept	24
Figure 13: CALM classic architecture [11]	25
Figure 14: OMNET++ simulation process	27
Figure 15: Architecture of NS-3 simulator [12]	28
Figure 16: DSRC layered architecture [13]	30
Figure 17: DSRC band plan [14].....	31
Figure 18: Time division between CCH and SCH interval [25]	32
Figure 19: WAVE Short Message format [15]	32
Figure 20: Veins simulation framework	34
Figure 21: State machine of the OMNET++ and SUMO communication modules [24].....	35
Figure 22: Sequence diagram of message exchange between OMNET++ and SUMO [24].....	36
Figure 23: SUMO simulation process	38
Figure 24: Layered architecture implementation	40
Figure 25: IEEE802.11p and MAC1609.4 parameters definition	41
Figure 26: Car ned file	42
Figure 27: RSU ned file	42
Figure 28: File structure of the omnet++ project.....	44
Figure 29: Sequence diagram of communication	45
Figure 30: Part of the general road scenario	46
Figure 31: Part of icy road scenario.....	46
Figure 32: Part of accident road scenario	47
Figure 33: General simulation	49
Figure 34: Icy simulation	50
Figure 35: Accident simulation.....	50
Figure 36: General scenario result evaluation	52
Figure 37: Minimum speed of vehicle1.....	53
Figure 38: Maximum speed of vehicle1	53
Figure 39: Icy scenario result evaluation normal direction.....	54
Figure 40: Icy scenario result evaluation reverse direction	54
Figure 41: Accident scenario result evaluation	55
Figure 42: Passenger cars statistics	63
Figure 43: OMNET++ simulation scenario selection	63
Figure 44: List of WSM WAVE Element ID [21]	64

10. References

- [1] Frötscher, Alexander; Fuchs, Susanne. Co-operative Networks for Intelligent Road Safety. COOPERS services and value chains, concerning operator/ user behaviour, Integration of services in the co-operative system (1.0). ATE, Hitec, January 2010. <http://www.coopers-ip.eu/fileadmin/results/deliverables/D3600-3700_COOPERS_SERVICES_AND_VALUE_CHAINS_VO4_APPROVED.pdf>, p.99
- [2] Frötscher, Alexander; Fuchs, Susanne. Co-operative Networks for Intelligent Road Safety. COOPERS services and value chains, concerning operator/ user behaviour, Integration of services in the co-operative system (1.0). ATE, Hitec, January 2010. <http://www.coopers-ip.eu/fileadmin/results/deliverables/D3600-3700_COOPERS_SERVICES_AND_VALUE_CHAINS_VO4_APPROVED.pdf>, p.154
- [3] Frötscher, Alexander; Faisstnauer, Christian. Co-operative Networks for Intelligent Road Safety. Final report on demonstration (1.0). AustriaTech, Autostrada del Brennero, June 2010. <http://www.coopers-ip.eu/fileadmin/results/deliverables/D6100_-_Final_report_on_demonstrations_V1_0.pdf>, p.29
- [4] Andreas Schmid et al. High Level Architecture. Architecture and System Specifications (2.0). CVIS, May 2009. <http://www.cvisproject.org/download/Deliverables/20090331_DEL_CVIS_3.2_High_Level_Architecture_v2.0.pdf>, p.9
- [5] Samson Tsegay et al. CVIS Test Site description and Planning (1.0). CVIS, April 2009. <http://www.cvisproject.org/download/Deliverables/DEL_CVIS_5.1_TS_Description%20and%20Planning_v1.0.pdf>, p.69
- [6] Samson Tsegay et al. CVIS Test Site description and Planning (1.0). CVIS, April 2009. <http://www.cvisproject.org/download/Deliverables/DEL_CVIS_5.1_TS_Description%20and%20Planning_v1.0.pdf>, p.71
- [7] Samson Tsegay et al. Test Site Results (1.0). CVIS, July 2010. <http://www.cvisproject.org/download/Deliverables/DEL_CVIS_5.2_Test%20Site%20Results_V1.0.pdf>, p.68
- [8] Roberto Brignolo et al. Description of the integrated vehicle and infrastructure platform. Horizontal Activities (2.0). CRF et al, April 2008. <http://www.safespot-eu.org/documents/D8.4.2_Integrated_vehicle_Infrastructure_Platforms.pdf>, p.39

- [9] Angela Spence et al. Final Report: Needs and Requirements of Infrastructure-based Sensing – Part A. INFRASTRUCTURE PLATFORM (7.0). MIZAR et al, Novembre 2006. <http://www.safespot-eu.org/documents/D2.2.2_Part_A_Needs_and_Requirements_of_infrastructure-based_sensing.pdf>, p.88
- [10] P. J. Feenstra et al. Test and validation plan. Cooperative Safety Systems Infrastructure Based (1.7). TNO et al, August 2009. <http://www.safespot-eu.org/documents/SF_D5.5.1_Test_and_validation_plan_v1.7.pdf>, p.30
- [11] Williams, Bob. The CALM Handbook, Continuous Air-interface Long and Medium range (1.2). September 2004. <http://www.tih.org.uk/images/c/c7/the_calm_handbook.pdf>, p.29
- [12] Network Simulator (NS-3), <https://www.nsnam.org/overview/key-technologies>
- [13] B. Kenney, John. Dedicated Short-Range Communications (DSRC) Standards in the United States (99.7). IEEE, July 2011, p.1164
- [14] B. Kenney, John. Dedicated Short-Range Communications (DSRC) Standards in the United States (99.7). IEEE, July 2011, p.1167
- [15] B. Kenney, John. Dedicated Short-Range Communications (DSRC) Standards in the United States (99.7). IEEE, July 2011, p.1172
- [16] Vodel, Matthias et al. SimANet – A Large Scalable, Distributed Simulation Framework for Ambient Networks (3). Technische Universität Chemnitz, December 2008. <<http://ojs.academypublisher.com/index.php/jcm/article/view/03071119/222>>
- [17] Pfliegl, R. COOPERS project presentation. AustriaTech, May 2005. <http://www.coopers-ip.eu/fileadmin/results/deliverables/D1-A1100_COOPERS_Project_Presentation_052010.pdf>, p.4
- [18] Kompfner, Paul. CVIS project presentation. ERTICO - ITS Europe. <http://www.cvisproject.org/download/cvis_project_presentation_del_10.pdf>
- [19] Kompfner, Paul et al. Final Activity Report. ERTICO - ITS Europe, August 2010. <http://www.cvisproject.org/download/Deliverables/DEL_CVIS_1.3_FinalActivityReport_PartII_PublishableSummary_V1.0.pdf>
- [20] Roberto, Brignolo. Co-operative Systems for Road Safety, Smart Vehicles on Smart Roads, February 2010. <http://docbox.etsi.org/Workshop/2009/200902_ITSWORKSHOP/CRF_BRIGNOLO_SAFESPOT.P>

DF>

- [21]Kurihara, Thomas M. et al. IEEE Standard for Wireless Access in Vehicular Environments (WAVE) - Networking Services. IEEE, December 2010.
<<http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5680697>>, p.119
- [22]Vilain, D. Et al. Test and validation results – ANNEX (1.4). June 2010.<http://www.safespot-eu.org/documents/SF_D5.5.2_Test_and_validation_results_ANNEX_v1.4.pdf>, p.23
- [23]Vilain, D. Et al. Test and validation results – ANNEX (1.4). June 2010.
<http://www.safespot-eu.org/documents/SF_D5.5.2_Test_and_validation_results_ANNEX_v1.4.pdf>, p.32
- [24]Sommer, Christoph et al. Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis (10). IEEE Transactions on Mobile Computing, January 2011.
<<http://www.ccs-labs.org/bib/sommer2011bidirectionally/sommer2011bidirectionally.pdf>>, p.6
- [25]B. Kenney, John. Dedicated Short-Range Communications (DSRC) Standards in the United States (99.7). IEEE, July 2011, p.1171
- [26]B. Kenney, John. Dedicated Short-Range Communications (DSRC) Standards in the United States (99.7). IEEE, July 2011, p.1164-1165
- [27]Ertekin,Ö et al. Test bench for I2V interfaces, including test vehicle, testing environment and test database (2.1). DOR, ARCS, ARCS et al., February 2010. < http://www.coopers-ip.eu/fileadmin/results/deliverables/D7-5300-5400-5600_TESTBENCH_VEHICLE_ENVIRONMENT_DATABASE_V2.1.pdf>, p.110-111

11. Appendices

Appendix A: Passenger car statistics

The following statistics shows the number of cars per 1000 people in different years. Each color represents a country except the light gray which represents European Union.

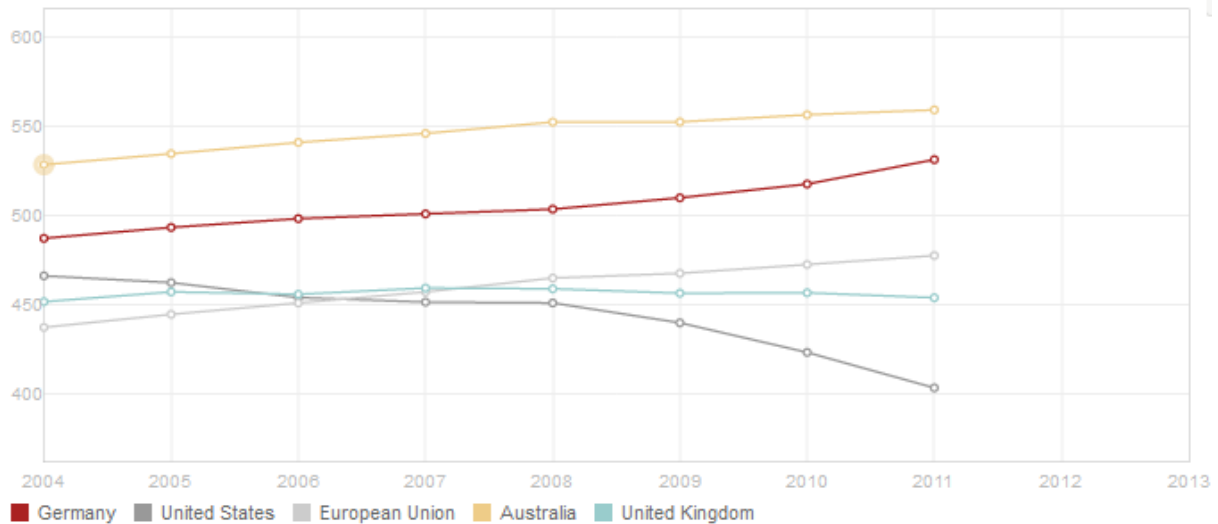


Figure 42: Passenger cars statistics

Source: <http://data.worldbank.org/indicator/IS.VEH.PCAR.P3/countries/DE-US-EU-AU-GB?display=graph>

Appendix B: OMNET++ Scenario selection

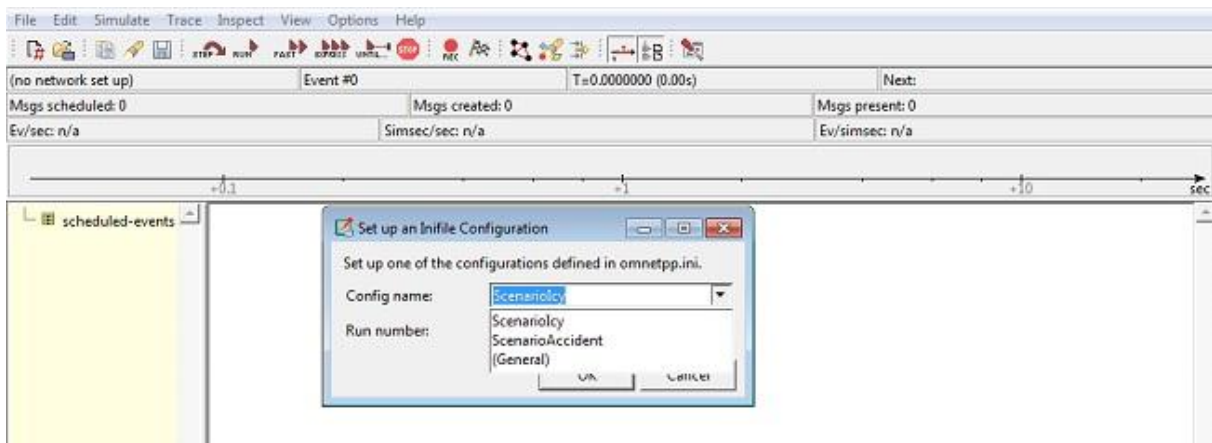


Figure 43: OMNET++ simulation scenario selection

Appendix C: Allocated WSM WAVE Element ID

In the Figure 44 the green rectangle area indicates the WSMP WAVE elements

	Element ID value	Used in	Reference
WSA WAVE Elements			
	0	Reserved	
Service Info	1	WSA Service Info	8.2.1
Channel Info	2	WSA Channel Info	8.2.1
WRA	3	WSA WRA	8.2.1
Extension WAVE Elements			
Transmit power used	4	WSA header, WSMP header	8.2.2.4, 8.3.4
2D Location	5	WSA header	8.2.2.4
3DLocationAndConfidence	6	WSA header	8.2.2.4
Advertiser identifier	7	WSA header	8.2.2.4
Provider Service Context	8	WSA Service Info	8.2.3.6
IPv6 address	9	WSA Service Info	8.2.3.6
Service port	10	WSA Service Info	8.2.3.6
Provider MAC address	11	WSA Service Info	8.2.3.6
EDCA Parameter Set	12	WSA Channel Info	8.2.5.8
Secondary DNS	13	WSA WRA	8.2.5.8
Gateway MAC Address	14	WSA WRA	8.2.5.8
Channel Number	15	WSMP header	8.3.4
DataRate	16	WSMP header	8.3.4
Repeat Rate	17	WSA header	8.2.2.4
Country String	18	WSA header	8.2.2.4
RCPI Threshold	19	WSA Service Info	8.2.3.6
WSA Count Threshold	20	WSA Service Info	8.2.3.6
Channel Access	21	WSA Channel Info	8.2.4.7
WSA Count Threshold Interval	22	WSA Service Info	8.2.3.6
	23 to 127	Reserved	
WSMP WAVE Elements			
WAVE Short Message	128	WSMP header	5.5, 7.3.2
WSMP-S	129	WSMP header	Annex F
WSMP identity supplement (WSMP-I)	130	WSMP header	IEEE P1609.1
	131 to 255	Reserved	

Figure 44: List of WSM WAVE Element ID [21]