Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Monterrey

School of Engineering and Sciences

NetMovExt: a dynamic extensor

A thesis presented by

Edson Jacobo Armenta Gastelum

Submitted to the
School of Engineering and Sciences
in partial fulfillment of the requirements for the degree of

Master of Science

in

Electronic Systems
(Telecommunications)

Monterrey Nuevo León May 15th, 2018

Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Monterrey

School of Engineering and Sciences

The committee members, hereby, certify that have read the thesis presented by Edson Jacobo Armenta Gastelum and that it is fully adequate in scope and quality as a partial requirement for the degree of Master of Science in Electronic Systems (Telecommunications)

Dr. Cesar Vargas-Rosales
Tecnológico de Monterrey
School of Engineering and Sciences
Principal Advisor

Dr. Leyre Azpilicueta Fernández de las Heras
Tecnológico de Monterrey
Committee Member

Tecnológico de Monterrey
11 MAY 2018
Dirección Nacional de Posgrado
Dr. Rubén Morales Menéndez

Dr. Alberto Francisco Martínez Herrera
Tecnológico de Monterrey
Committee Member

Dr. Rubén Morales Menéndez
Dean of Graduate Studies
School of Engineering and Sciences

Monterrey Nuevo León, May 15th, 2018

Declaration of Authorship

I, Edson Jacobo Armenta Gastelum, declare that this thesis titled, 'NetMovExt: a dynamic extensor" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Edson Jacobo Armenta Gastelum
Monterrey Nuevo León, May 15th, 2018

Dedication

Dedico este trabajo a cada una de las personas que me apoyaron de manera incondicional. En primer lugar, agradezco a Dios por permitirme tener vida y salud para alcanzar esta meta. Amigo San José de Cupertino (protector de los estudiantes) fuiste mi fiel guardián en este proyecto, a ti ¡GRACIAS! Fueron varias las personas que indudablemente me otorgaron su apoyo. Inicio agradeciendo al Dr. Cesar Vargas-Rosales por su apoyo y su confianza en mí, para la realización de este trabajo. A Carolina Zapata Hernández que me regaló su tiempo, su atención y su conocimiento, gracias amiga mía. A Miguelys Rodríguez Barreto, mi comadre quien me otorga su amistad incondicionalmente y me abrió un espacio en su hogar y en su familia. Agradezco a mis padres y hermana por sus palabras y su actitud motivante frente a la vida. A mi esposa Johanna Christine Hahn-Armenta que me alentó, motivó a retomar este trabajo y no desistir, valoro y atesoro su amor, optimismo y su positivismo para ver la vida, gracias mi amor. A mi hija que fue la inspiración más significativa, pues su existencia en mi vida me exigió sin duda demostrarme que soy capaz. Mi intención como padre de enseñar con el ejemplo, me requirió congruencia en mis actos, obteniendo la autoridad moral para criar, desarrollar y motivar a mi hija Luise Emelie Armenta Hahn y a cualquiera de mis hijos, a lograr lo que se propongan: Fuerza, Inteligencia y Corazón. A mi hermano parte esencial de este trabajo quien me otorgo su paciencia, su dedicación, su esfuerzo, su tiempo, su apoyo y su amor para lograrlo, sin duda el pilar de este trabajo.

¡GRACIAS A CADA UNO DE USTEDES!

Acknowledgements

Dedico este trabajo a cada una de las personas que me apoyaron de manera incondicional. En primer lugar, agradezco a Dios por permitirme tener vida y salud para alcanzar esta meta. Amigo San José de Cupertino (protector de los estudiantes) fuiste mi fiel guardián en este proyecto, a ti ¡GRACIAS! Fueron varias las personas que indudablemente me otorgaron su apoyo. Inicio agradeciendo al Dr. Cesar Vargas-Rosales por su apoyo y su confianza en mí, para la realización de este trabajo. A Carolina Zapata Hernández que me regaló su tiempo, su atención y su conocimiento, gracias amiga mía. A Miguelys Rodríguez Barreto, mi comadre quien me otorga su amistad incondicionalmente y me abrió un espacio en su hogar y en su familia. Agradezco a mis padres y hermana por sus palabras y su actitud motivante frente a la vida. A mi esposa Johanna Christine Hahn-Armenta que me alentó, motivó a retomar este trabajo y no desistir, valoro y atesoro su amor, optimismo y su positivismo para ver la vida, gracias mi amor. A mi hija que fue la inspiración más significativa, pues su existencia en mi vida me exigió sin duda demostrarme que soy capaz. Mi intención como padre de enseñar con el ejemplo, me requirió congruencia en mis actos, obteniendo la autoridad moral para criar, desarrollar y motivar a mi hija Luise Emelie Armenta Hahn y a cualquiera de mis hijos, a lograr lo que se propongan: Fuerza, Inteligencia y Corazón. A mi hermano parte esencial de este trabajo quien me otorgo su paciencia, su dedicación, su esfuerzo, su tiempo, su apoyo y su amor para lograrlo, sin duda el pilar de este trabajo.

¡GRACIAS A CADA UNO DE USTEDES!

NetMovExt

by

Edson Jacobo Armenta Gastelum

Abstract

Nowadays, there is still an open field in network simulators dedicated to working VANETs. They have been introduced to add some networking characteristics by acting as adapters or extension between mobility tool and network module such as ASH, OVNIS, VEINS, VnetIntSim, TraN, GrooveSim. In this thesis, we propose a platform focused on VANETs atmosphere. This contemporary simulator works with two types of communication: Vehicular to Vehicular (V2V) in a multiple-hop scheme and Vehicular to Infrastructure to Vehicular (V2I) in a single-hop scheme. We have aimed our work to have a flexible architecture for networking side where other services such as modules can be added in parallel. In fact of that, we integrated a module with our own developed testing routing protocol based on geographical position. Additionally, we included a beta propagation module to show flexibility of structure. A customizable GUI was integrated for easy interaction with user.

Contents

Chapter 1

Introduction

Advances in mobile computing and wireless communications in transportation systems has led to the rise of new lines of research in the field of wireless communications and the availability of new applications and services that were not possible before. Wireless communication between vehicles has enabled safety applications such as accident avoidance and traffic congestion alerts, [1]. Assistance for drivers and safety applications can feed from traffic information to help users avoid congestion via warning messages, [2].

Many of these advances have become possible thanks to the rise of Vehicular Ad-hoc Networks (VANETs) as the enabling technology. As great interest in Vehicle-to-Vehicle (V2V) communications has grown rapidly, [3]. There is also Vehicle-to-Infrastructure (V2I) communications that make use of fixed communication infrastructure like a Road-Side Unit (RSU) and a Traffic Control Center. RSUs and Traffic Control Centers can work as information centers that collect useful information about road characteristics from vehicles. Vehicles may adjust their traveling routes and RSUs may inform the traffic control center. This information can then be broadcasted to other vehicles for drivers to take decisions based upon it. Therefore, investigating how to utilize the collected data can result onto a range of other useful applications, [4].

The collected data from Simulators helps to evaluate new applications. Simulators play an important role in searching for solutions to real environments. Tests using live infrastructure is not feasible because of extensive and expensive deployment. Not only vehicular traffic simulations, but also network layer simulations can be added jointly to perform an effective evaluation of inter-vehicular communications on issues such as routing mechanisms, safety methods and travel experience. In a wireless network, a network simulator monitors the vehicles as nodes and simulates data dissemination among them through data transfer emulation, [5].

1.1 Motivation

The latest research has been focused on Intelligent Transportation System (ITS) because advanced technologies have been integrated into transportation infrastructure and vehicles, [6]. VANETs are considered a key technology for ITS to improve road safety and comfort. These systems aim to provide services, like traffic management, to vehicles to be better informed and make safer, more coordinated, and smarter use of transport networks. Recently, VANETs has turned in a very popular topic for research in wireless networking and in automotive industries, [7].

The establishment of communication links between moving vehicles carries new problems and challenges. In VANETs, high speed plays an important factor. Speed makes standard network protocols inefficient or unusable, [2]. In this field, the routing protocols have an important task to do. They must find the best choice of ways to communicate vehicle to vehicle, [1]. Network simulation is a powerful tool to evaluate the efficiency and reliability of new protocols for wireless environment scenarios, or proposed networking solutions. This is vital prior to implementing or deploying networks due to its low cost and reduced complexity, [5].

Most models have dealt with the effect of mobility, but only a few simulate the impact of vehicular communications, [8]. For VANETs, there is still a requirement to develop new wireless communication standards, and routing protocols are still a complex challenging issue, [9]. Therefore, communication network simulators are a non-complex way of assessing applications as well as protocols, [10]. Simulations represent a significant part to evaluate approaches as physical test beds are difficult to implement, [11].

1.2 Problem Statement and Context

As new applications are developed and extended, VANETs become an important subject of research. However, due to the high cost of deploying and implementing VANET systems in a real environment, most research focuses on computer simulations, [12]. Nonetheless, despite the many efforts to provide a simulation framework with mobility models that reflect real environments and scenarios as closely as possible, there is still a work to be done, [13].

Some examples can be in applications like road safety, traffic avoidance as well as their performance accuracy when compared to real scenarios. Besides, some studies have been made to develop more efficient protocols for on-demand route planning according to observed traffic congestion or incidents as well as for safety applications. Realistic simulation of network protocol behavior in VANETs scenarios is strongly demanded for evaluating the applicability of newly developed network protocols, [13].

A realistic simulation can help to predict and assess the impact of different parameters of communications as protocols over VANET services. There are some simulators focused on conventional networks as NS or OMNET++. They provide support to simulate TCP and other protocols. The most recent version offers wireless channel modeling for LANs and others. There are few pieces of works that are able to offer a dedicated environment to test exclusively VANETs inter-communication, [14].

Simulation frameworks are an ongoing attempt to understand the dependence between vehicular traffic and communications performance, [15]. This has led to the creation of models that try to provide a suitable environment for the simulation and evaluation of ad hoc communication performance, [16]. Simulators usually support small-scale vehicular networks unlike large population in ITS, [8].

Thus, the simulation of large-scale systems will require implementing different methods of simulation. Moreover, various platforms have been able to model real-time vehicular mobility and inter-vehicular communication within different scenarios both traffic and road conditions. However, the majority of current tools are focusing on modeling the impact of mobility on communications, while a few works on the effect of inter vehicular communications over vehicular mobility, [8].

1.3 Research Question

As previously discussed, there are some simulating platforms which connect networking to mobility for VANETs. One problem to face in VANETs is the capacity of simulators to support large-scale urban environments. In fact of that, simulations focus particularly on the impact of mobility on vehicular communications under different traffic and roadway scenarios.

Scalability is a significant characteristic that contemporary platforms must take into account. This is becoming a starting point to developing on current researches. How to increase abilities and capacities of extensions that connects network part to vehicular mobility for VANETs is still an actual open field. However, the impact from intercommunication to mobility is also a main part of the current interest of research.

Additionally, simulators with a wide variety of networking simulation characteristics yield simulations with more realistic results. Versatility conducts platforms to expand abilities on simulations. Besides, sequential simulation lacks the processing resources to simulate urban transportation networks in real time. Contemporary platforms can be enhanced to have meaningful results.

1.4 Objectives

General Objective:

To integrate a dynamic platform where networking models and vehicular mobility approaches work together in an easily scalable and versatile environments.

Objective 1:
To develop a network core with routing, delivery mechanism and events generator.

Objective 2:
To include routing and propagation modules as part of modular structure.

1.5 Methodology

It is well-known there are several simulators for wireless communications. Some examples include NS, and OMNeT++, and extensors like ASH, OVNIS, VEINS, VnetIntSim, TraN and GrooveSim. Until now, there is a need for simulation specialized tools that considers communication factors, mobility and scalability. In this thesis we offer an extensible, flexible, scalable and dedicated platform with some state-of-the-art factors in VANETs simulators that leads to network testing.

We have added a generator of random events such as accidents. Such events test the platform capacity to handle and manage communication resources. We have seen that message delivery may become a significant factor to evaluate the networking mechanism. Moreover, it is possible to add more complexity to our scenarios through the addition of two types of communication schemes (V2V, VI2IV) as well as the user configuration for parameters like transmission range, number of RSUs and so on.

It is assumed that each vehicle has an On-Board Unit (OBU), and the existence of Road-Side Units (RSUs) along the road that eventually communicate with vehicles. For this research, the OBUs and RSUs resources are assumed to be unlimited to simplify the work. Besides, V2V communications for vehicles follow a Multi-hop scheme while Single-hop scheme is used for V2I. The nodes use the current geographical location of the neighbors by Global Positioning System (GPS). This information is quite important to choose the adequate next hop for data forwarding.

In VANETs, the radio propagation conditions rapidly change. It is known that both V2V and V2I channel models are required. It is important to have a deep understanding of wireless propagation channels to be able to have efficient communication systems that provide fast information access. A deep channel modeling, data loss, and communication delay are out of the scope, [17]. By implementing a vehicular mobility model, the testing scenarios can represent the closest realistic traffic phenomena such as traffic jams, high and low density of cars and lack of cars, and so on.

We have taken into account a simple propagation phenomenon. We have integrated a beta propagation module to add loss of power of the wireless signal. To keep the work simple, we assume that there are no obstacles that could interfere with the communication as a clear visibility scheme. It is assumed that a bidirectional channel is working to exchange information dynamically and simultaneously both between nodes and RSUs. By this mean, vehicles that have been informed are assumed to be able to avoid accidents by changing lanes.

Transmission and delivery have been simulated however data packet content modeling in messages and security mechanism are out of scope. We assumed there are no errors in packet delivery unless there are no vehicles in range to pass on the message. Likewise, it is assumed that both OBUs and RSUs use radio over Wireless Access in Vehicular Environment (WAVE) with 5.9 GHz and Media Access Control MAC layer meets the requirements to work properly. Physical and MAC layer in detail are out of scope.

For the networking module, we have developed and implemented a basic position-based routing protocol. We proposed a two-phase routing mechanism. The first phase uses Dijkstra to get all possible destinations from a single source and then builds a graph of those group of nodes for every time step. The second part applies the chosen algorithm depending on the module selected by the user, which decides the best route to deliver the message.

Our routine uses a modified depth-first approach available in the NetworkX library for Python to generate all simple paths from a specific source to all possible targets. It does not contain repeating paths or permutations. It then chooses the first path in the list that contains the least number of nodes.

In addition, to prove flexibility of expansion, we added a beta propagation module. This module provides other variables for routing decisions and delivery. Moreover, by activating the propagation module along with the routing module, the probability of choosing a certain path would depend on the combination of both. Therefore, the platform might make the final decision through only routing protocol, propagation or the cooperation criteria.

This simulator is capable of simulating scenarios with V2V which have been used to satisfy the interaction among vehicles and V2I to enable fixed infrastructure to vehicles communication. The V2V communication scheme is always active by default. Some parameters of the networking environment can be set through the GUI. A mobility pattern is the initial input on a text file. Prior to the network simulation, the mobility trace is generated by using VanetMobiSim.

1.6 Main contributions

This research provides a scalable and module-structure platform for VANETs. This tool contains a mobility model of VANETs. The platform can work with V2V and V2I through multi-hop and single-hop delivery methods. Our work can manage from small, to large-scale transportation simulation. It has a flexible composition where more routing services can be added as modules.

In addition, platform is able generate random events as means to test routing protocol efficiency. Our tool is able to put events in any time during simulations. In this way, it opens a chance to simulate behavior of protocols with a variety of traffic at one time. We have included two modules with 3 different metrics to select routes. There are routing, propagation and cooperation criteria.

In order to add more realistic environments, we added a propagation approach through a second module. A simple propagation criterion where the distance between 2 nodes is equal to a probability of a link being establishes. A short distance means a high probability; in contrast longer, distances will have a low probability. Definitely, a visual representation of nodes can help to compare easily paths resulted from protocols therefore we included path-layout generation feature.

Now, our work can read a mobility external file with number of cars, car ID, step, coordinates and speed. For the moment, the platform generates statistics with the networking behavior obtained from the simulation, by this means; to plan a network can be feasible. On the other hand, a GUI has been included to set and manage network parameters of simulation.

In chapter two, our research reviews and describes a brief of platforms and the mobility simulator VanetMobiSim. Then in chapter three, we introduced our work in detail. The global characteristics of the simulator are described. Then, the geographical-base protocol is presented. Afterwards, propagation and cooperation modules, their metrics and criteria are explained. Then, a description of forwarding and delivery processes. After that, the interface GUI is shown as well as its fields and parameters. Moreover, in chapter four, we review testing scenarios and their results. Finally, in chapter five, we have our conclusions and future works.

Chapter 2

Simulating Platforms

This chapter gives a general overview of extension platforms for network simulators. It contains general characteristics, features and general functionality of those tools. In VANETs, some platforms have been developed as extensions whose objective is not only join mobility and networking but also create or modify dynamically trajectories. Each tool works with a specific network platform and mobility simulator. On the other hand, this section provides an overview of VanetMobiSim which is the vehicular mobility model used on this research.

2.1 Extension Network Simulators

.

Simulators are commonly used to evaluate new VANET applications and communication protocols. The simulators that evaluate exclusively network communication process are shown in Figure 2.1.1. Recently, extensions that can connect simultaneously external mobility simulator and network platforms have been developed. They are shown in Figure 2.1.2. In addition, they added features to increase capabilities. Therefore, we have briefly described the most popular extensors similar to our research platforms.

Network Simulators
NS-3
NS-2
SWANS
OMNET++
OPNET
QUALNET

Figure 2.1.1 shows various network simulators. They are some of wireless network simulators which are developed for conventional networks.

Dynamic Platforms		
Tool	**Mobility Simulator**	**Network Simulator**
ASH	IDM/MOBIL, IVG	SWANS
OVNIS	SUMO	NS-3
STRAW	STRAW	SWANS
VEINS	SUMO, IVC	OMNET++
VnetIntSim	INTEGRATION	OPNET
TraNS	SUMO	NS-2
iTETRIS	SUMO	NS-3
Automesh	Customizable to add any mobility model	NS-2 or Qualnet

Figure 2.1.2 shows a variety of extensors for network simulators. They are the dynamic platforms whose work is to join network part to mobility model and other features.

2.1.1 Application-aware SWANS with high mobility (ASH)

It is an extension of the network model SWANS which provide an application aware model which works bidirectional communication. In addition. ASH uses IDM module to model how cars follow others as well as Minimizing Overall Braking decelerations Induced by Lane Changes (MOBIL) module for Lane-change management. ASH also works with Inter-Vehicle Geocast (IVG) and probabilistic IVG (p-IVG) protocol of message broadcasting. ASH uses SWANs network model. Among main characteristics of AHS are, [8]:

- Modeling two-way communication between the mobility and networking models
- Modeling highway topology
- Modeling mobility states
- Intelligent broadcast
- Logging and statistical facilities

2.2.2 Online Vehicular Network Integrated Simulation (OVNIS)

This Platform is a realistic manager of vehicles capable of adjusting node mobility and generating vehicular traces simultaneously. This extension works with SUMO and NS-3 and controls an interconnection between them. OVNIS counts on a vehicular traces generator. Moreover, it has a traffic aware network Manager which takes care of interconnection between traffic simulator and nodes application modules. This platform uses NS-3 model for network environment, [8].

2.1.3 Street Random Waypoint (STRAW)

This simulator uses its own vehicular model and can create road segments, intersections, traffic control mechanism. It can model individual vehicles as nodes, including high speed and inter-vehicular communication. STRAW can set vehicle properties including maximum speed, reaction time and acceleration rate, road segments. At each intersection, traffic control mechanism grants deterministic admission control protocols for vehicles. Its architecture consists of 3 interacting component models: Intra e inter-segment mobility model and a route management and execution model. It can be integrated into any wireless network simulator as SWANS, [8].

2.1.4 Vehicles in Network Simulation (VEINS)

The Vehicles in Network Simulation (VEINS) can evaluate IVC protocol impact on road traffic mobility. This hybrid framework consists of a network and a road traffic simulator. Besides, to support control between network and vehicular simulator, it has a communication channel. This platform uses OMNET++ as its network simulator. Dedicated modules are used to support communication. On the other hand, for vehicular simulation uses SUMO. It receives commands from OMNET++ through mobility traces are generated. OMNET++ uses theses traces to modify routes and speed of vehicles. By using Veins, protocols can be evaluated both V2V and V2I, [8].

2.1.5 Vehicular Network Integrated Simulator (VNetIntSim)

This platform has modules to link INTEGRATION traffic simulator with communication network platform OPNET by a bidirectional channel. Upon VNetIntSim starts execution, the channel gets established. Then, both simulator exchanges initial messages and they get synchronized. OPNET receives location of each vehicle which is sent and calculated from INTEGRATION. Then, vehicles information becomes updated by one of linking modules. It can simulate simple V2V and V2I scenarios, [8].

2.1.6 Traffic and Network Simulation Environment (TraNS)

In this platform Traffic and Network Simulation Environment is 2 modes of simulation. By network centric simulation traffic flow is generated i.e. music or travel information. The traffic simulator generates a simulation trace and the network simulator the trace file. In application-centric simulation, extraordinary events can be generated e.g. collision avoidance and abrupt braking. Sumo is used as mobility simulator and NS-3 as network platform, [8].

2.1.7 Integrated Wireless and Traffic Platform for Real-Time Road Traffic Management Solutions (iTETRIS)

It is a large-scale ITS applications simulator for vehicular networks. It supports WiMAX, UMTS, DVB-H wireless and radio access technologies as well as ETSI. It has a modular composition and supports external modules. It is known that this platform performs accurate and complex simulations for realistic traffic scenarios. iTETRIS accepts files from SUMO which interact with NS-3. Moreover, some features were added as fuel consumption information, speed and route changes suggestion beside traffic congestion, [8].

2.1.8 Automesh

This ITS-application simulation framework works with both QUALNET or NS-2 as network model. Automesh has 5 modules in its structure. It has a module for inter-vehicle communication which receives the data from driving simulator to modify routes. It generates a dynamic mobility model for each vehicle that supports speed limits, traffic signals, acceleration and deceleration. Propagation module takes charge of evaluation for correctness and performance of communication protocols. By geographic database module, geographic information becomes available. Finally, it provides a Graphical user interface to configure and playing simulations. Automesh offers 3 plug-in modules to take care of vehicle control, propagation and communication control, [8].

2.1.9 GrooveSim

It works with its own communication model and traffic vehicular model. The author has included the Groovenet routing protocol. It is a hop-based communication protocol with a dedicated short-range communication. The communication part models a two-state Gilbert Elliot Markov, a collision and a channel model to assure concurrent communication. By using a message diffusion mode, the communication protocol works to periodically exchange congestion information. For critical information, it uses message directed mode to immediately report alert messages. The mobility model based on number of vehicles, road segments speed limits minimum and maximum speed and a Markovian four-state probabilistic model determines vehicular mobility, [8].

2.2 VANET Mobility Simulator (VanetMobiSim)

The performance of network protocols is affected by connectivity dynamics when they are used in large-scale communication systems. The high-speed, driving rules-constrain, and road topology-limited mobility are unique characteristics on vehicle-based networks. VanetMobiSim is one of the most realistic and configurable tool for the generation of vehicular motion in the category of isolated models therefore, the car following model, an intersection management and multilane capability demonstrated that they are the minimum requirements for a realistic modeling of vehicular motion, [18].

VanetMobiSim considers the selection of the road topology. The length of streets, he frequency of intersections, the density of buildings can significantly impact mobility metrics. VanetMobiSim allows to define the road topology by four ways: (a) user-defined graphic (XY coordinates define by user), (b) GDP map (city map imported .GDF restricted files), (c) TIGER map (city map imported from TIGER data base only for USA), (d) Clustered Voronoi graph (randomly generated by Voronoi tessellation). They are shown in Fig 2.2.1. In this way, this platform can emulate urban and rural scenarios, [18].

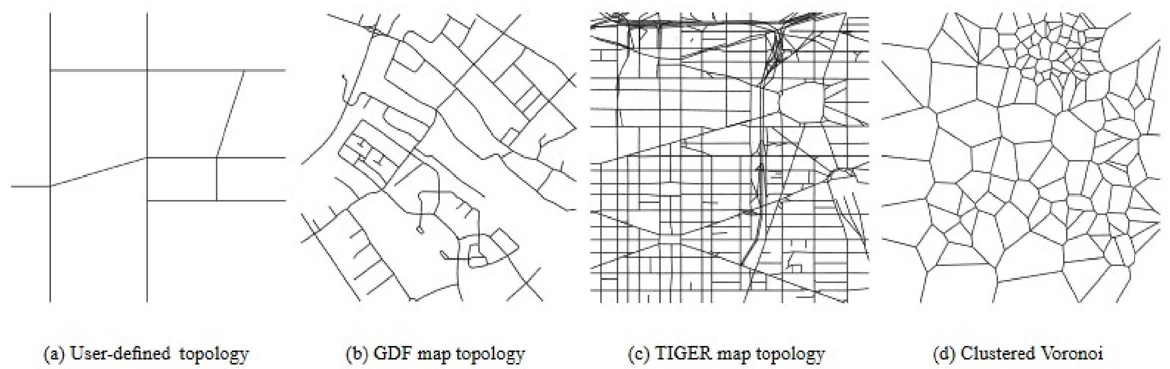


Fig. 2.2.2. The four simulation areas which VanetMobiSim supports as source for simulations.

Mobility models which belong to car-to-car category provide a reliable image of real world vehicular dynamics. The Intelligent Driver Model (IDM) comes from the car-following category. At first, the IDM-IM arose and brought with him the intersection handling to the car-to-car interaction by IDM. It can manage crossroads controlled by stop signs. Secondly, the IDM-LC, it added overtaking capability to vehicles and is based on the MOBIL lane changing model, [2].

VanetMobiSim works based on CanuMobiSim which it is a simulator coded in Java and generates mobility traces. VanetMobiSim adds the vehicular mobility support a higher degree of realism. VanetMobiSim produces detailed vehicular movement traces with a high degree of realism. VanetMobiSim consists mainly on a vehicular spatial model which is composed of spatial elements, [19].

VanetMobiSim takes into account intersection handling using stop signs and traffic lights, it allows vehicles to change lanes to overtake each other and introduces multi lane roads [19]. It makes uses of Intelligent Driving Model with Intersection Management (IDM-IM) and Intelligent Driving Model with Lane Change (IDM-LC), [2]. The IDM-LC through VanetMobiSim was used to simulate vehicular mobility along with realistic user's behavior and topology traits.

A user-defined model uses a set of vertices and edges composing the backbone of the vehicular spatial model. This vehicular oriented model supports microscopic level mobility model named IDM-LC describing perfectly car-to-car, intersection managements and overtaking model (MOBIL). Overtaking features interact with IDM-IM to manage lane changes and vehicle accelerations and decelerations. The IDM-LC model supports parameters speed, vehicle's length, maximal acceleration and deceleration, and Intersection management, [20].

In addition to the previous enhancements, macro and micro mobility features are taken into account. Our study took in advantages of those two since we describe a road topology called User-defined graph, stop signs, and a trip generator module that defines the sets of points or interest and also a path computation module which compute the best path among points. This last feature works by three different ways to select the best way to reach the destination. The first runs a Dijkstra's algorithm, the second not only considers the length, but also traffic jam levels and the last one takes the two previous ways plus the road speed limit. As mentioned, micro-mobility traits are included as well.

Aspects related to an individual car's speed, acceleration, smooth speed variation, car queues, traffic jams and over takings. Note that all of them give more realism to vehicular movements. Moreover, mobility models are in the latter features however this study works with only one of them, IDM-LC model. It computes individual car's speed based on the presence or nearby vehicles interacting with intersection management and overtaking features. This section only provides a general description of the mobility tool, specific information please refer to [2] and [19].

Chapter 3

NetMovExt

This chapter contains the main parts of the platform we proposed. It gives a whole description of our research. We have considered 5 important points to describe in detail. One, global characteristics where a general overview is described. Two, geographical based protocol, where we have proposed a simple protocol to test routing mechanism. Three, we present a beta propagation module and cooperation criteria. Four, the forwarding and delivery processes are introduced. And finally, the GUI is shown and described. These chapters sections work as the main reference of details mentioned throughout this thesis.

3.1 Global Characteristics

To develop our research, we have used Python. It is an open-source programming language. We chose python because of its ease of use, wide support in the scientific community and rapid prototyping capabilities that allowed us to focus on the content other than the tool. It has been used in multiple fields as: education, engineering and scientific and numerical computing and so on. It provides a friendly environment to users ranging from beginners to experienced programmers. There is widely available online support information, code examples, recipes and patterns on official https://www.python.org/.

We have implemented the routing module following an extensible design based on inheritance principles of Object Oriented Programming. By implementing a parent class module as an Abstract Base Class (ABS), we provide an enforcing template for new modules being implemented. Thus, every module must inherit from parent class module and implement all three abstract methods that are assumed as a contract for it to work seamlessly with the rest of the system.

The core is composed by two primary modules. The geographical protocol module works with geographical location to select paths. And a propagation module which works with a simple propagation "beta" method. Then, a feature as well which combines both criteria to make a decision for routing. We need a mobility pattern as input for our platform, we have used primarily variables in the text output file from VanetMobiSim.

Mobility generator gets its information from a customizing XML file. Simulation is created through a simulating area in which nodes are set and specified by a mobility model. Time simulation is given by number of steps and roads are built with vertex coordinates. Moreover, speed, and other important parameters must be turned on. Once the input file is run, a summary file is generated in the same directory in which the input file is located. This file contains the result from output with autos, RSUs, and flow of messages: source and destination.

The core receives the input and queries the current routing module for the selected path chosen according to the algorithm. Then the control returns to the core that uses this output path to carry on with the execution of the simulation. Figure 3.1.1 shows the structure of simulator.

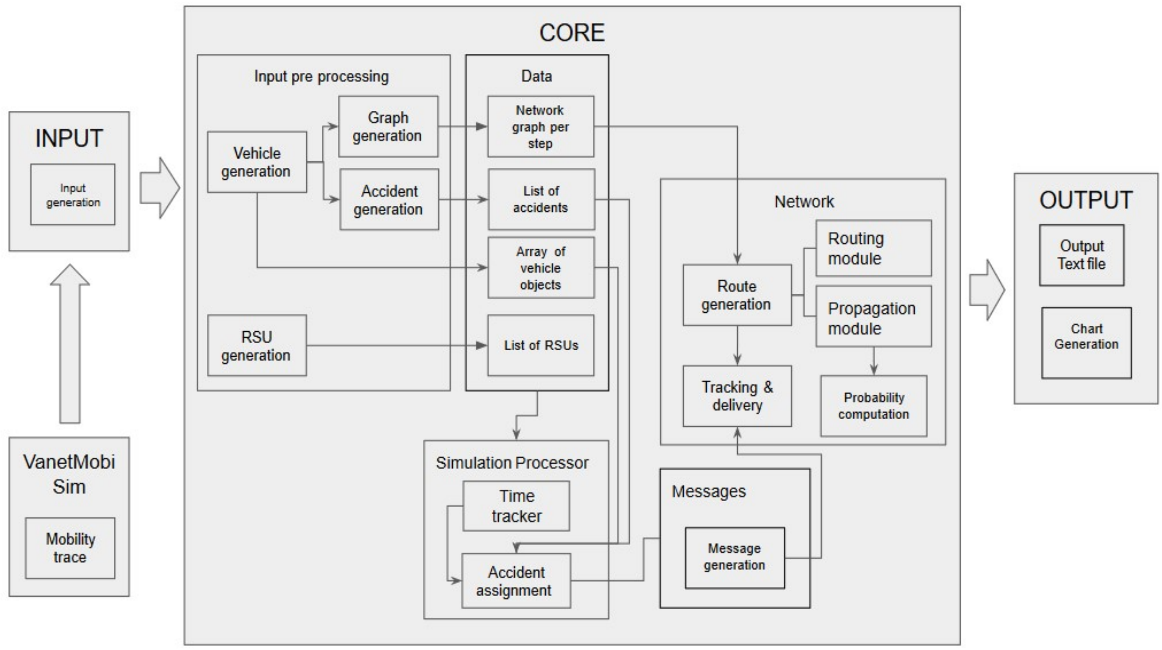


Figure 3.1.1 shows the layout of simulator. As input, networking parameters and mobility trace. In the core, pre-processing, data, simulation processor, network model. As output, a text file and charts

Figure 3.1.2 shows how the routine works with external data and internal process.

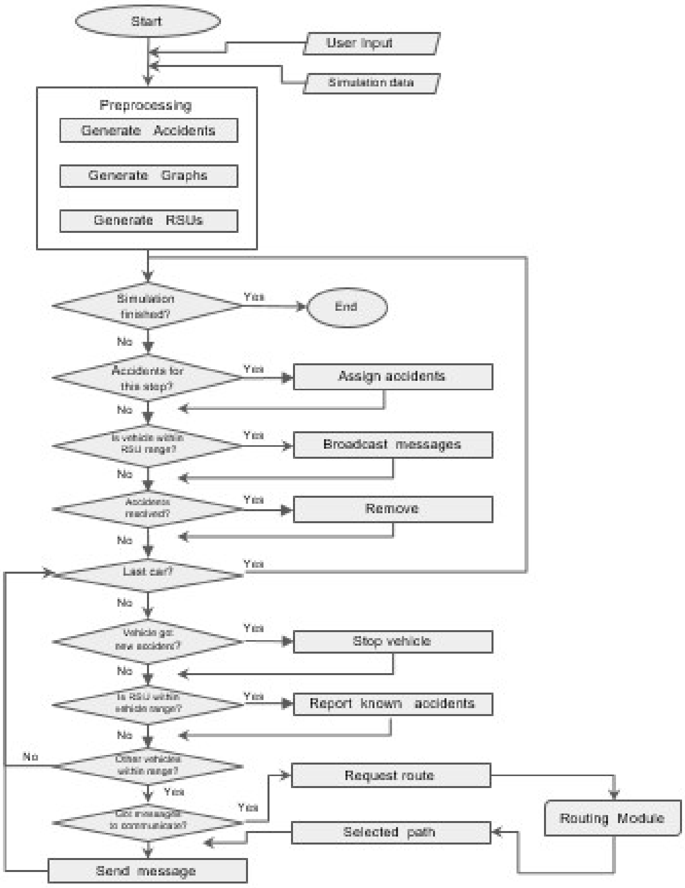


Figure 3.1.2 shows the flowchart. It contains the main routine to make decisions

A GUI was integrated with customizable parameters to set a network. Through the GUI, platform can be controlled, there are buttons for RUN, RESET and QUIT. Additionally, it can provide statistic chart, graphs of nodes, RSU coordinates. The way to use it is shown in figure. 3.1.3.

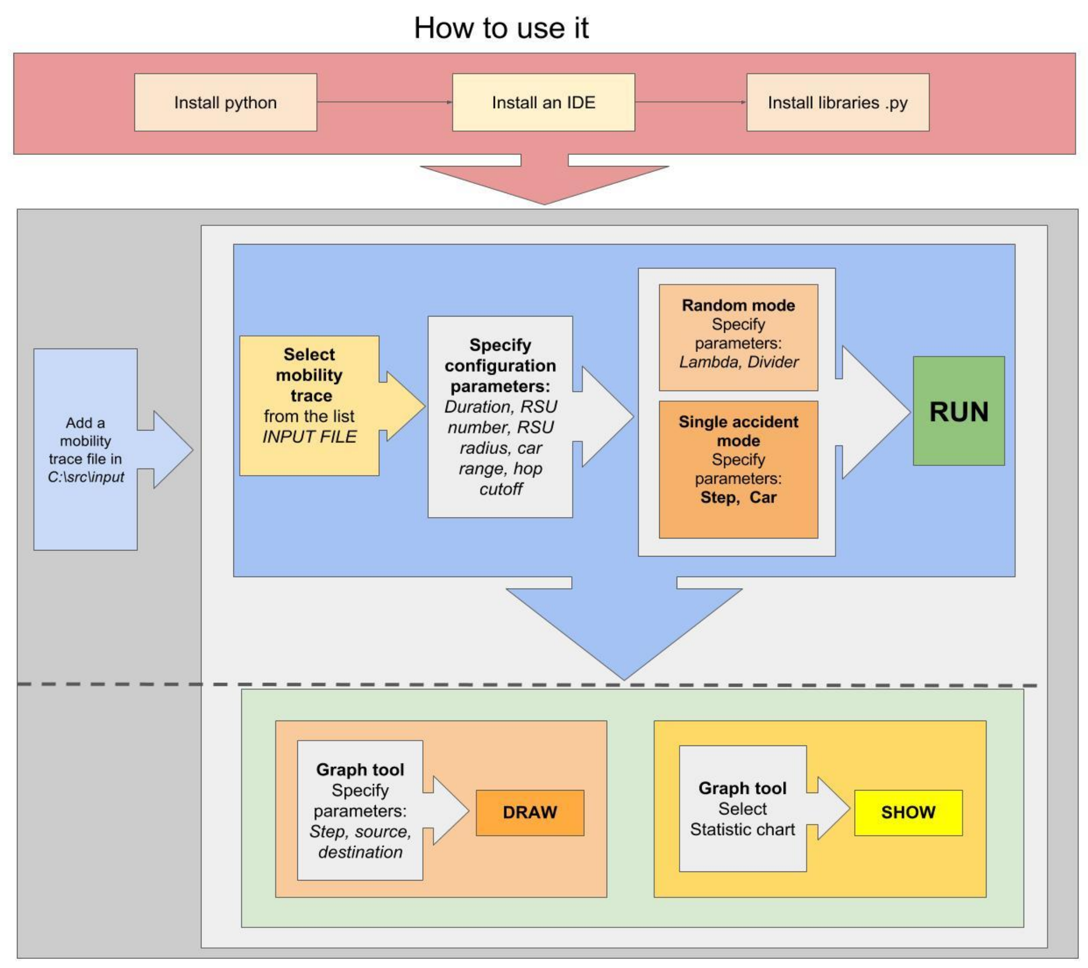


Figure 3.1.3 shows how to use the tool, from top to bottom and right to left.

Our tool can display out different information through the output sections. It consists of 2 sections. The first shows a general output. It can display available cars per step, accident per step, RSU status coverage, sending node and receiving node. Second screen trace output, shows up information similarly to number one but it shows only specific cars previously selected. Also, platform has a default graph display to show statistics. It has been defined to display out message per step, messages per RSU.

We have proposed generating events such as the occurrence of accidents. They are random events that can occur multiple times throughout the simulation. With accidents, routing can be tested. We used an exponential distribution to generate the time between accidents with λ within the preprocessing stage.

An accident happens at a specific spot and lane to any car even if the speed $v = \frac{m}{s}$ is zero. Accidents have a certain lifetime $t = s$ in steps. They are assumed to have been resolved by the end of their lifetime. Only one accident can occur to any specific car at any particular time.

There is always a same direction lane available for vehicles to take over and keep their way without stopping. In other words, at the moment cars involved in accidents do not affect lane transit. Cars are assumed to have the same power radio range $r_c = m$. A car antenna is at the center above the roof. The propagation is assumed to spread with equal power in all directions. When a new vehicle enters the radio range r_c of an informed one, it receives the warning and becomes informed.

For V2V scheme communication, we have worked with emergency messages dissemination. These messages contain accident warnings which travel and inform other cars. A car can send and receive any number of messages within its range. We proposed routing protocols that use “the fewest hops” or “short distance, higher probability” as criteria to spread emergency messages. The former protocol is based on geographical location of vehicles and the latter uses distance between nodes to calculate a probability. There is always a GPS active onboard.

For V2I method, RSUs were included in a median strip along the road. To keep things simple, we have limited the amount of possible RSUs along the analyzed area. They can be placed only if the distance between their coverage does not overlap each other. If one of the cars reaches an RSU, it passes on the message that contains the existence of the accident. Then the RSU will send the message to all cars in its coverage.

In Vehicles, OBUs are in charge of all communications and computation tasks, while GPS works to acquire geographical position. As we stated previously, there is an infinite amount of network resources for communication units and infrastructure; so that reception, processing and retransmission of messages can be done simultaneously at any scale regardless of the number of vehicles involved.

In a future stage, a significant feature to include is a dedicated bidirectional link between mobility model and network part. This contribution will provide route planning control on demand by changing routes dynamically. As mentioned before, other services can be included by adding more modules. A more complex propagation service, a model of data in packets or a security scheme module. Likewise, many protocols to test can be developed and tested.

3.2 Core

3.2.1 Geographical-based Protocol Module

Before our main processing routine, we designed a pre-process stage or "discovery process", where we go through every time step and build a graph with nodes as cars and edges as geographic distance of cars within transmission range, all remaining disconnected nodes are removed.

We created an initial route table with edges among nodes that are within the parameterized transmission range. We bring back the system graph for every time step with the corresponding system information and update them to the corresponding step inside the initial route table. In this way, we save computation time by avoiding the "discovery stage" at the beginning of each time step.

Therefore, at every time step inside the main routine, we retrieve the corresponding graph from the initial route table to run the routing process only if required by an accident occurrence. In this way, the routing process starts with identifying corresponding neighbors to the source node from the information saved in the table. Later, if an accident occurs both modules can find a path and update the table.

As mentioned above, we proposed a geographical-position based protocol to route data to prove functionality. It selects the best path with "fewest-hops" criterion. Basically, it was developed as following:

1. It uses Dijkstra's algorithm to gather all possible node destinations for a single selected source node.
2. Secondly, it uses an optimized version of Depth First Search (DFS) to find all combinatorial simple paths.
3. Thirdly, it calculates the size of each path and returns the shortest one.
4. Once that the best path is selected, the mechanism delivers the message to a target.

DFS uses a modified depth first approach available in the NetworkX library. This resource generates all simple paths from a specific source to all non-repeating possible targets. The graph is formed only with a group of nodes in transmission range of any of those vehicles.

3.2.2 Propagation Module and Cooperation

A significant part of VANETs networking simulation is undoubtedly propagation phenomena. Our research includes a simple propagation module “beta” to offer a more realistic environment. Besides, by the second module addition, flexible structure is shown. We have proposed a scheme with probabilities for single edges among nodes.

The distance is proportionally inverse to the probability of link establishment between two nodes. That means that longer distances yield lower probabilities to establish a connection hindering message forwarding. Thus, the routing mechanism will prefer taking the highest probability link between two nodes to route the message. That will help to ensure message delivery to the next node.

As the result for both modules working together, the system gives a weighted combination of probability and number of hops. Then, the delivery mechanism currently uses weights of 80% for propagation criterion and the rest 20% for “fewest hop” criterion. For the cooperation criteria, we established the following simple equation that produces higher values for shorter routes and lower values for the longer ones, Figure 3.2.1. Finally, the system will pick the path with the higher value result.

$$f(x) = 0.8 * p(x) + 0.2 * (1 - \frac{l(x)}{l_{max}})$$

X = route
p(x) = probability
l(x) = length in hops
l(max) = maximum length in hops

Figure 3.2.1 shows the cooperation criteria equation. To combine both propagation and routing criterion, this equation was formulated.

Summarizing, the final delivery mechanism works with three possible criteria.

1. Based on the routing module working with “the fewest-hops” criteria to route and deliver message.

2. Based on the propagation module working “with the highest probability on shortest distance” criteria for delivering.

3. Based on cooperation between propagation and routing modules. This mechanism weights highest probability criterion with 80% and the fewest-hop criterion with 20%. The final decision is the highest grade as result of the cooperation of both modules.

The highest probability in third criterion has always higher priority over the fewest-hops criterion. This resolution method works always if both modules are active at the same time. In case that the propagation module was the only one active the result would be exclusively the highest probability. In other words, the path with the shortest total distance among the possible paths.

3.2.3 Forwarding and delivery

In V2V, emergency messages must be transmitted on high priority from vehicle to vehicles with a multi-hop scheme. At the time of the event, the affected car sends emergency messages to in range cars passing by.

The conditions for V2V in multi-hop retransmission are:

1. Cars retransmit to any other car that comes within range and hasn’t been informed.
2. An accident can be transmitted, forwarded and delivered only if its life is not over yet.
3. Cars in opposite direction can communicate each other as long as they are in range.
4. A car involved in an accident cannot receive its own warning from others.
5. The number of hops is fixed to 1 as default value.

For V2I, the single hop scheme does not use any routing mechanism. Each vehicle gives the RSUs the message when it is within car’s range r_c. In a similar fashion, when a car enters to the RSU coverage r_r, it receives a message. The RSU does not know which car drops a message. The tool can position and locate RSUs gathering the geographical location and distributing them to avoid overlapping areas.

In some cases, warnings can reach the RSUs depending on whether there is a least one car in which the message can go through. That vehicle must be the closest one at least r_c from that RSU. Then, RSU will broadcast the message to all cars within its transmission range $r_r - m$ including the car which delivered the message in RSU.

3.3 GUI

The GUI holds four components in which one is to input values and the other three display different information, see Figure 3.3.1.

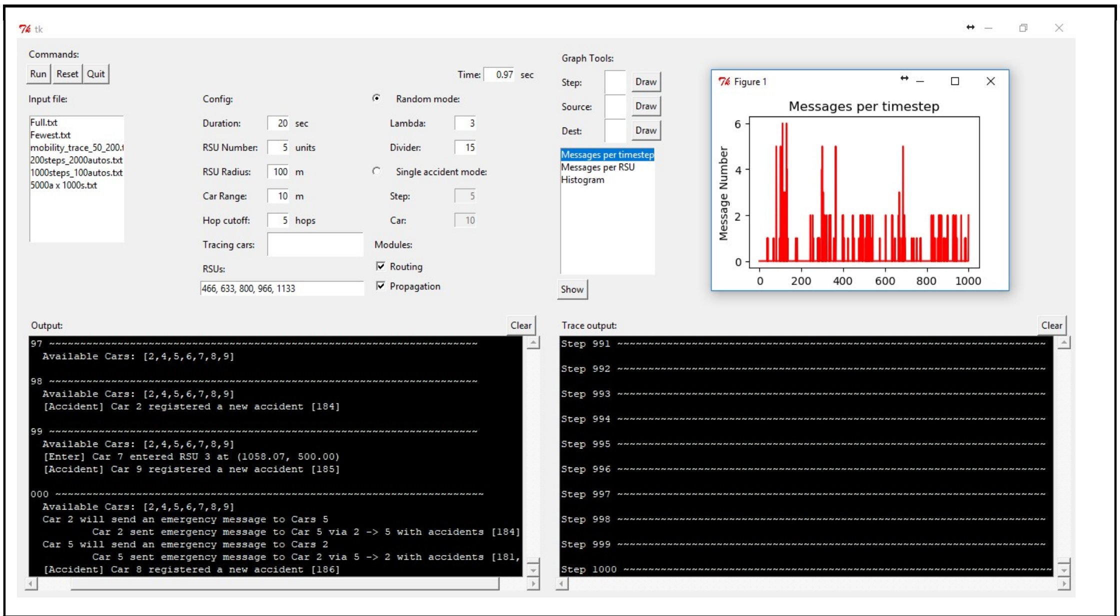


Figure 3.3.1 shows the main panel of the GUI. Some parameters are customizable.

1. First Quadrant holds several text fields to introduce initial values:
    a. Commands:
        i. Run: To play simulation
        ii. Reset: To clear text fields and graphs
        iii. Quit: To close simulator
    b. Input file: To select the initial text file
    c. Configuration: To set simulation parameters
        i. Duration: To set accident duration
        ii. RSU Number: To set the amount for RSU
        iii. RSU Range: To set the radio of coverage
        iv. Car Range: To set the antenna car coverage
        v. Hop cutoff: To set the maximum number of hops
        vi. Tracing cars: To set the auto Id to trace
        vii. RSUs: To show the center of location for RSUs
    d. Random Mode: To set accident generation

i. Lambda: To set the number of events
ii. Divider: To set the interval of time for each Lambda

e. Single Accident Mode: To set an isolate generation
   i. Step: To set the isolate step to monitor
   ii. Car: To set the isolate car to monitor
f. Modules
   i. Routing: To set criteria of fewest hops
   ii. Propagation: To set criteria of shortest distance
   iii. Both: to set criteria of 80-20%
g. Time: To display computational time.

2. Second Quadrant: It displays the whole output of simulation
3. Third Quadrant: It display the specific output
4. Fourth Quadrant: To select and display result charts
   a. Graph Tools: To set specific step, source and destination as well as "draw" to get node graph:
      i. Step: To display the node graph on a specific step
      ii. Source: to display the graph for a specific node
      iii. Destination: to display the graph for a specific destination

Chapter 4

Results

We analyzed how the cars behaved as we tested against different population sizes and simulation length. Several situations may be found in a city from low to high density in different hours. Our selected scenarios emulated the closest situation to the reality of urban environment in a city. Then, we conducted simulations to demonstrate functionality and performance of our platform.

During the simulation, vehicles move in-road according to the IDM-LC model. Vehicles are uniformly generated and distributed at the vertices of the roads. They have maximum and minimum speed $v_{max} = \frac{m}{s}$ and $v_{min} = \frac{m}{s}$. The vehicles can accelerate $a = \frac{m}{s^2}$ and decelerate $b = \frac{m}{s^2}$ to their speed. Parameters as time, population and others were set up in VanetMobiSim. Vehicular parameters are in the Figure 4.1

General Simulation Parameters		
Type of node	Car	
Traffic model	IDM-LC	
MAC Layer	WAVE	
Speed	Min	Max
	6.3m/s	12.5m/s
Car´s Length	5m	
Acceleration (a)	$0.6m/s^2$	
Deceleration (b)	$0.9m/s^2$	
Jam distance (s0)	2m	
Safe time headway (t)	1.5s	
Step	0.1s	
Stay	0s	
Politeness factor (p)	0.5	
Acceleration threshold (athr)	$0.2m/s^2$	
Mobility Simulator	VanetMobiSim	

Figure 4.1 shows the vehicular and networking parameters for test-bed scenarios.

For our proposed scenarios, we had one road along 1 km with two lanes per direction. Scenarios with a large XY coordinates area can be simulated by setting "simulation area" on VanetMobiSim. The size of area depending on the road topology source. We used User-defined graph nevertheless our platform must be modified depending on which source was used. Our vehicles are running along road l_r by a determined time t. We have observed different behaviors as much as a variety of cars and time. The proposed time and population of cars were based on observed behaviors according to distance.

Vehicles run along on the evaluation area of our analysis shown in the figure as a red square. When vehicles reach the end, they turn back on the opposite direction of the road. The endpoints of the road and their immediate neighboring area in the graphic representation were discarded to simplify the work. The blue line represents the track line where RSUs can be placed. A graphical representation is shown in Figure 4.2.

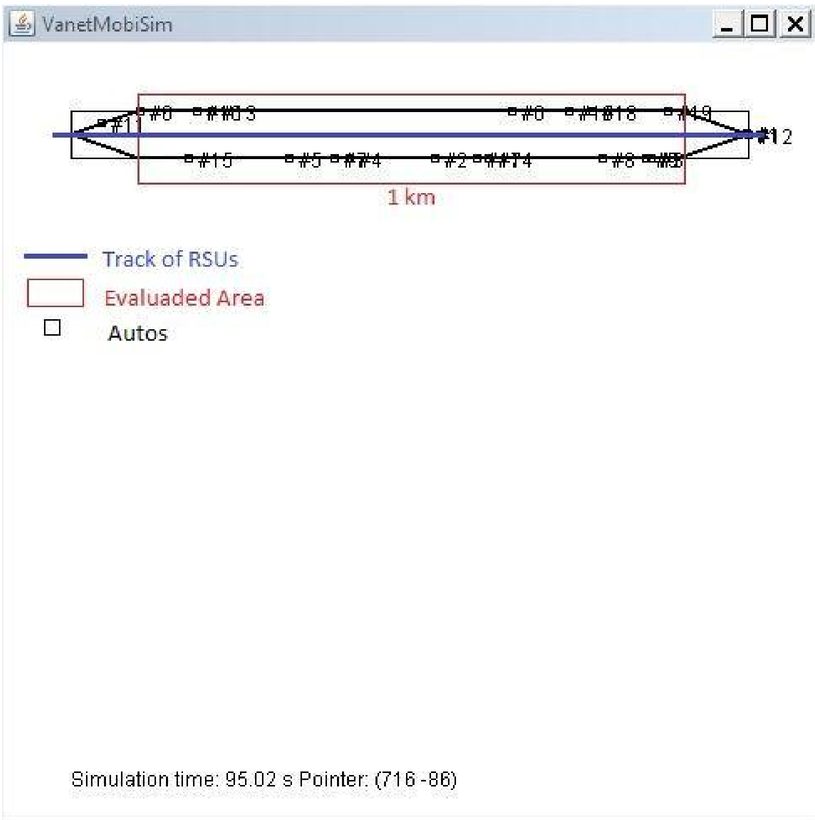


Figure 4.2. Shows a view of mobility simulation. One road with two lanes per direction. The red rectangle shows the evaluated area, the blue line represents where to place RSUs and the smalls square are vehicles.

We considered 1000 s as a maximum standard base time t_s for a sufficient interaction among cars. Regarding cars, we emulated “traffic jam” by observing the number of cars in l_r . A “high density” of cars represents a quick interaction among cars in an instant. Thus, we considered 2000 cars as “high density” as well as “low density” with 100 autos. For V2V, we have a default and minimum value of hops $h_m = 1$. And for V2I, there were no overlapping areas between RSUs coverage. All the parameters are shown in Table 4.3

Simulaton Parameters			
Distance $lr = m$	1000		
Time $t = s$	200	1000	
Modulos	Routing	Propagation	Cooperation
Lambda ratio	3/15		
Duration $da = s$	20		
RSU number R	0	5	
RSU Radius $rr = m$	100		
Car Range $rc = m$	10		
Hops hb	5		
Minimun Hops hm	1		

Figure 4.3 shows the simulation parameters for scenarios.

Sometimes, we included two or more scenarios in one test. We consider a scenario with “low density” of cars at t_s in l_r. The last interesting scenario considered traffic “high density” and traffic jam. Our platform has customizable features like car transmission range, amount of RSUs and the routing modules enabled among others. All the testing scenarios are in the following Figure 4.4

Scenarios						
One straight road 1km - 2 lanes x direction - Both directions						
Scenario	Time	Autos	RSUs (r=100m)		R	P
			0	5	on	on
1	1000s	100	☑		☑	
2	1000s	100	☑			☑
3	1000s	100		☑	☑	
4	1000s	100		☑		☑
5	1000s	100		☑	☑	☑
Fewest	1000s	10		☑	☑	☑
Full	1000s	2000		☑	☑	☑

Table 4.4 shows the testing scenarios proposed to prove functionality and performance.

For all testing scenarios, we have evaluated schemes of communication as “networking challenges”. Besides V2V and V2I, we have called V2I to describe the direction of transmission from vehicle to infrastructure. In this way, we can analyze if the platform works with a method or more for communication.

Moreover, we firstly ran each scenario with RANDOM MODE to discover where many messages took place. It means that vehicles formed a network. Once we chose a car in that network, we went secondly on SINGLE MODE to put specifically an accident on that isolated car in previously discovered step.

4.1 Basic Functionality Scenario

The front panel of the GUI is shown in Figure 4.1.1 Only V2V occurred due to there were no RSUs for this scenario.

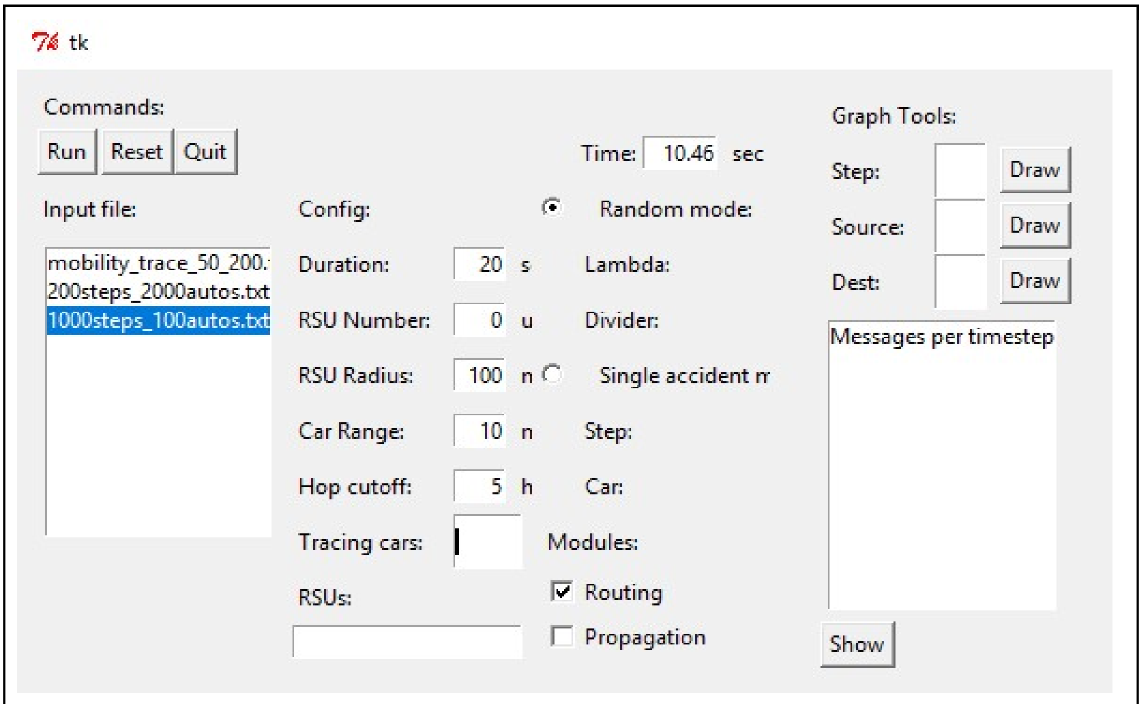
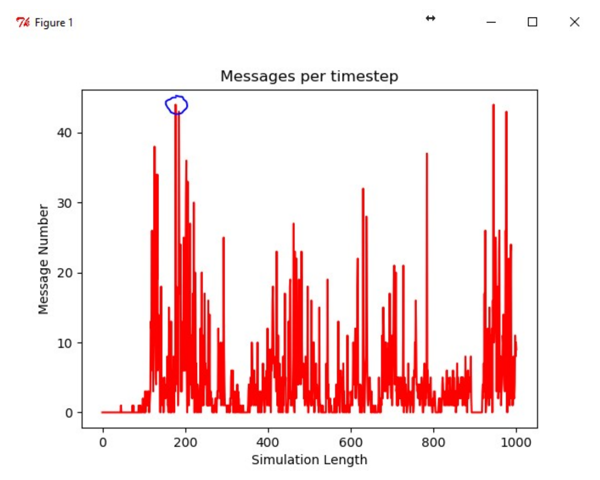
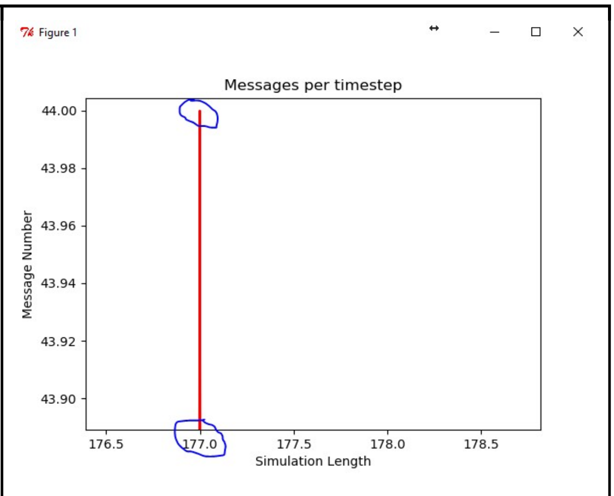


Figure 4.1.1 shows the control panel of GUI. It contains all the options to set simulations.

In Figure 4.1.2a, it can be observed how the transmissions behaved in the whole simulation on RANDOM MODE. The Figure 4.1.2b shows a closeup of the moment in which the first highest number of hops took place, it was in step 177 with 44 retransmissions.



4.1.2a shows a chart for the first scenario in random mode for the whole simulation



4.1.2b shows a closeup of the first simulation for the highest value.

The output screen displays the process of the whole simulation in RANDOM MODE. In Figure 4.1.3 we can observe that car 29 began the V2V transmission with 28 to Cars 33, 55, 58. However, the transmission to car 58 in red failed because of h_b. On the other hand, green lines show succeeded transmissions as car 47 with the accidents 28 and 29 to Cars 71, 72, 73, 48, 49, 50, 52, 54, 29, 31. Upon happening accident 31, the car 47 sent the message to Cars 99 and 93 in the same step.

Step 177 ~~~~~~
Available Cars:
[6,7,9,13,17,18,19,22,23,24,25,26,28,29,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,84,85,86,88,89,90,93,94,95,96,97,99]

Car 29 will send an emergency message to Cars 33, 55, 58

Car 29 could not reach Cars 58

Car 29 sent emergency message to Car 33 via 29 -> 73 -> 71 -> 52 -> 55 -> 33 with accidents [28]

Car 29 sent emergency message to Car 55 via 29 -> 73 -> 71 -> 52 -> 55 with accidents [28]

Car 29 sent emergency message to Car 58 via with accidents [28]

Car 33 will send an emergency message to Cars 71, 72, 73, 48, 49, 50, 52, 54, 29, 31

Car 33 sent emergency message to Car 71 via 33 -> 55 -> 52 -> 71 with accidents [28, 29]

Car 33 sent emergency message to Car 72 via 33 -> 55 -> 52 -> 71 -> 73 -> 72 with accidents [28, 29]

Car 33 sent emergency message to Car 73 via 33 -> 55 -> 52 -> 71 -> 73 with accidents [28, 29]

Car 33 sent emergency message to Car 48 via 33 -> 48 with accidents [28, 29]

Car 33 sent emergency message to Car 49 via 33 -> 55 -> 52 -> 71 -> 73 -> 49 with accidents [28, 29]

Car 33 sent emergency message to Car 50 via 33 -> 48 -> 50 with accidents [28, 29]

Car 33 sent emergency message to Car 52 via 33 -> 55 -> 52 with accidents [28, 29]

Car 33 sent emergency message to Car 54 via 33 -> 55 -> 52 -> 54 with accidents [28, 29]

Car 33 sent emergency message to Car 29 via 33 -> 55 -> 52 -> 71 -> 73 -> 29 with accidents [28, 29]
Car 33 sent emergency message to Car 31 via 33 -> 55 -> 52 -> 31 with accidents [28, 29]
[Accident] Car 47 registered a new accident [31]
Car 47 will send an emergency message to Cars 99, 93
Car 47 sent emergency message to Car 99 via 47 -> 99 with accidents [28, 31]
Car 47 sent emergency message to Car 93 via 47 -> 93 with accidents [28, 31]

Figure 4.1.3 show a partial view of the general output for Scenario 1 in the Figure 4.3.

As a tool, the platform can generate node graphs. In the following Figure 4.1.4, we could see the graphs for the node 33 which retransmitted both events 28 and 29.

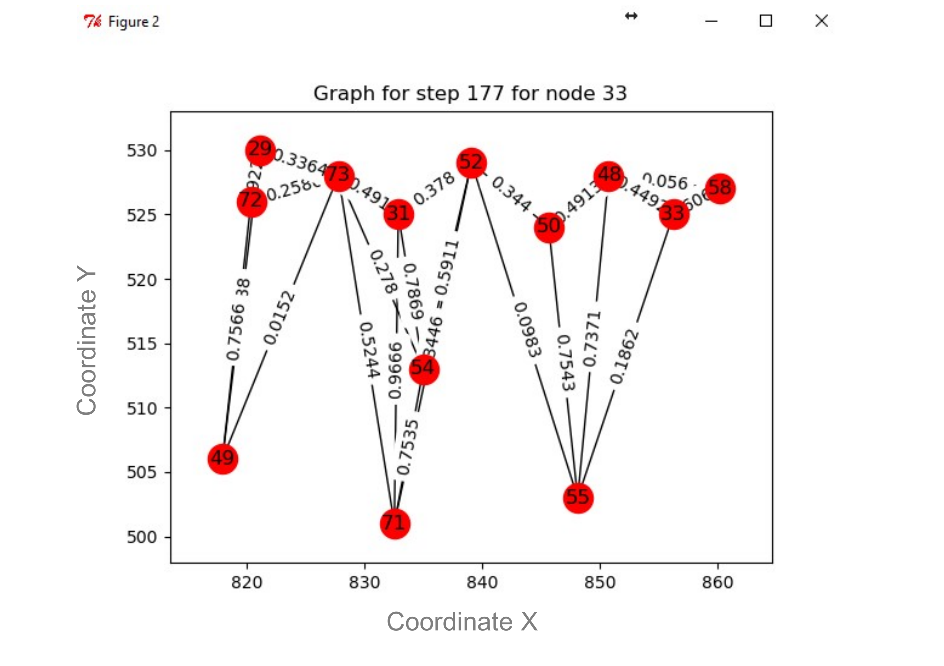


Figure 4.1.4 shows the view of the map of nodes in RANDOM MODE.

Moreover, if we used SINGLE ACCIDENT MODE to isolate the simulation for same car 33 from the 177 step, it shows an isolated result in the trace output. The result is shown in the Figure 4.1.5 below.

Step 177 ~~~~~~
[Accident] Car 33 registered a new accident [0]
Car 33 will send an emergency message to Cars 71, 72, 73, 48, 49, 50, 52, 54, 55, 58, 29, 31
Car 33 sent
emergency message to Car 71 via 33 -> 55 -> 52 -> 71 with accidents [0]
Car 33 sent
emergency message to Car 72 via 33 -> 55 -> 52 -> 71 -> 73 -> 72 with accidents [0]
Car 33 sent
emergency message to Car 73 via 33 -> 55 -> 52 -> 71 -> 73 with accidents [0]
Car 33 sent
emergency message to Car 48 via 33 -> 48 with accidents [0]
Car 33 sent
emergency message to Car 49 via 33 -> 55 -> 52 -> 71 -> 73 -> 49 with accidents [0]
Car 33 sent
emergency message to Car 50 via 33 -> 48 -> 50 with accidents [0]
Car 33 sent
emergency message to Car 52 via 33 -> 55 -> 52 with accidents [0]
Car 33 sent
emergency message to Car 54 via 33 -> 55 -> 52 -> 54 with accidents [0]
Car 33 sent
emergency message to Car 55 via 33 -> 55 with accidents [0]
Car 33 sent
emergency message to Car 58 via 33 -> 58 with accidents [0]
Car 33 sent
emergency message to Car 29 via 33 -> 55 -> 52 -> 71 -> 73 -> 29 with accidents [0]
Car 33 sent
emergency message to Car 31 via 33 -> 55 -> 52 -> 31 with accidents [0]

Step 178 ~~~~~~

Step 179 ~~~~~~
Car 33 will send an emergency message to Cars 37, 38
Car 33 sent
emergency message to Car 37 via 33 -> 58 -> 52 -> 50 -> 38 -> 37 with accidents [0]
Car 33 sent
emergency message to Car 38 via 33 -> 58 -> 52 -> 50 -> 38 with accidents [0]

Step 180 ~~~~~~

Step 181 ~~~~~~

Step 182 ~~~~~~
Car 33 will send an emergency message to Cars 36, 63
Car 33 could not reach Cars 36, 63
Car 33 sent
emergency message to Car 36 via with accidents [0]
Car 33 sent
emergency message to Car 63 via with accidents [0]

Figure 4.1.5 shows a partial output view using tracing cars and SINGLE ACCIDENT MODE.

4.2 Routing vs Propagation Scenario

For scenario 2, we took the same car 33 and using the same mobility pattern to run to get the results by SINGLE ACCIDENT MODE with routing module. The output is shown in Figure 4.2.1

```
Step 177 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    [Accident] Car 33 registered a new accident [0]
    Car 33 will send an emergency message to Cars 71, 72, 73, 48, 49, 50, 52, 54, 55, 58, 29, 31
                                                                      Car 33 sent
emergency message to Car 71 via 33 -> 48 -> 50 -> 52 -> 54 -> 71 with accidents [0]
                                                                      Car 33 sent
emergency message to Car 72 via 33 -> 55 -> 52 -> 31 -> 73 -> 72 with accidents [0]
                                                                      Car 33 sent
emergency message to Car 73 via 33 -> 48 -> 50 -> 52 -> 31 -> 73 with accidents [0]
                                                                      Car 33 sent
emergency message to Car 48 via 33 -> 48 with accidents [0]
                                                                      Car 33 sent
emergency message to Car 49 via 33 -> 55 -> 52 -> 31 -> 73 -> 49 with accidents [0]
                                                                      Car 33 sent
emergency message to Car 50 via 33 -> 48 -> 55 -> 50 with accidents [0]
                                                                      Car 33 sent
emergency message to Car 52 via 33 -> 48 -> 55 -> 50 -> 52 with accidents [0]
                                                                      Car 33 sent
emergency message to Car 54 via 33 -> 48 -> 55 -> 50 -> 52 -> 54 with accidents [0]
                                                                      Car 33 sent
emergency message to Car 55 via 33 -> 48 -> 55 with accidents [0]
                                                                      Car 33 sent
emergency message to Car 58 via 33 -> 58 with accidents [0]
                                                                      Car 33 sent
emergency message to Car 29 via 33 -> 55 -> 52 -> 31 -> 73 -> 29 with accidents [0]
                                                                      Car 33 sent
emergency message to Car 31 via 33 -> 48 -> 50 -> 52 -> 54 -> 31 with accidents [0]

Step 178 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Step 179 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    Car 33 will send an emergency message to Cars 37, 38
                                                                      Car 33 sent
emergency message to Car 37 via 33 -> 58 -> 52 -> 50 -> 38 -> 37 with accidents [0]
                                                                      Car 33 sent
emergency message to Car 38 via 33 -> 58 -> 52 -> 50 -> 48 -> 38 with accidents [0]

Step 180 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Step 181 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Step 182 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    Car 33 will send an emergency message to Cars 36, 63
    Car 33 could not reach Cars 36, 63
                                                                      Car 33 sent
emergency message to Car 36 via with accidents [0]
                                                                      Car 33 sent
emergency message to Car 63 via with accidents [0]
```

Figure 4.2.1 shows a part of the trace output in SINGLE ACCIDENT MODE with routing module.

By doing a comparison of both criteria, we show the results in Figure 4.2.2. In orange, we have the result for Propagation module and without color for Routing module. As we can notice in green, in some cases both criteria get the same results. This is likely to there is a unique path.

```
=
Step 177 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
[Accident] Car 33
registered a new accident [0]
Car 33 will send an
emergency message to Cars 71, 72, 73, 48, 49, 50, 52, 54, 55, 58, 29, 31
=
Car 33 sent
emergency message to Car 71 via 33 -> 48 -> 50 -> 52 -> 54 -> 71 with accidents [0]
Car 33 sent
emergency message to Car 71 via 33 -> 55 -> 52 -> 71 with accidents [0]
Car 33 sent
emergency message to Car 72 via 33 -> 55 -> 52 -> 31 -> 73 -> 72 with accidents [0]
Car 33 sent
emergency message to Car 72 via 33 -> 55 -> 52 -> 71 -> 73 -> 72 with accidents [0]
Car 33 sent
emergency message to Car 73 via 33 -> 48 -> 50 -> 52 -> 31 -> 73 with accidents [0]
Car 33 sent
emergency message to Car 73 via 33 -> 55 -> 52 -> 71 -> 73 with accidents [0]
=
Car 33 sent
emergency message to Car 48 via 33 -> 48 with accidents [0]
=
Car 33 sent
emergency message to Car 49 via 33 -> 55 -> 52 -> 31 -> 73 -> 49 with accidents [0]
Car 33 sent
emergency message to Car 49 via 33 -> 55 -> 52 -> 71 -> 73 -> 49 with accidents [0]
Car 33 sent
emergency message to Car 50 via 33 -> 48 -> 55 -> 50 with accidents [0]
Car 33 sent
emergency message to Car 50 via 33 -> 48 -> 50 with accidents [0]
Car 33 sent
emergency message to Car 52 via 33 -> 48 -> 55 -> 50 -> 52 with accidents [0]
Car 33 sent
emergency message to Car 52 via 33 -> 55 -> 52 with accidents [0]
Car 33 sent
emergency message to Car 54 via 33 -> 48 -> 55 -> 50 -> 52 -> 54 with accidents [0
Car 33 sent
emergency message to Car 54 via 33 -> 55 -> 52 -> 54 with accidents [0]
Car 33 sent
emergency message to Car 55 via 33 -> 48 -> 55 with accidents [0]
Car 33 sent
emergency message to Car 55 via 33 -> 55 with accidents [0]
=
Car 33 sent
emergency message to Car 58 via 33 -> 58 with accidents [0]
=
Car 33 sent
emergency message to Car 29 via 33 -> 55 -> 52 -> 31 -> 73 -> 29 with accidents [0
Car 33 sent
emergency message to Car 29 via 33 -> 55 -> 52 -> 71 -> 73 -> 29 with accidents [0
Car 33 sent
emergency message to Car 31 via 33 -> 48 -> 50 -> 52 -> 54 -> 31 with accidents [0
Car 33 sent
emergency message to Car 31 via 33 -> 55 -> 52 -> 31 with accidents [0]
=
  Step 178 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
  Step 179 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Car 33 will send an
emergency message to Cars 37, 38
Car 33 sent
emergency message to Car 37 via 33 -> 58 -> 52 -> 50 -> 38 -> 37 with accidents [0
=
Car 33 sent
emergency message to Car 38 via 33 -> 58 -> 52 -> 50 -> 48 -> 38 with accidents [0
Car 33 sent
emergency message to Car 38 via 33 -> 58 -> 52 -> 50 -> 38 with accidents [0]
```

=

Step 180 ~~~~~~

Step 181 ~~~~~~

Step 182 ~~~~~~

Car 33 will send an emergency message to Cars 36, 63

Car 33 could not reach Cars 36, 63

Car 33 sent emergency message to Car 36 via with accidents [0]

Car 33 sent emergency message to Car 63 via with accidents [0]

Step 183 ~~~~~~

Step 184 ~~~~~~

Figure 4.2.2 shows a comparison of routing criteria vs propagation criteria.

Now, we can see a comparison of isolated graph of node 33 to node 49. The Figure 4.2.3a and 4.2.3b show graphs for each criterion. We could notice that Propagation criterion resulted with different selected nodes unlike routing criterion result. Propagation works with distance between nodes and compares probability. Therefore, short distances by propagation may cost more hops or take different nodes that routing criterion.

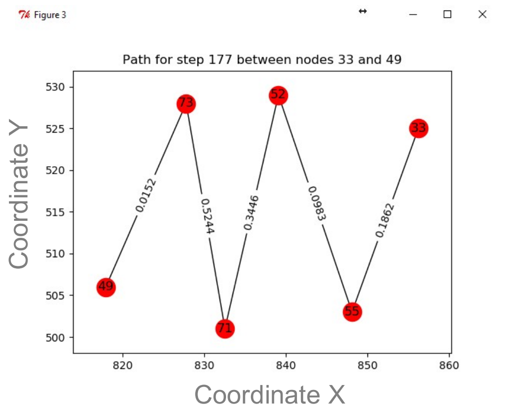


Figure 4.2.3a shows a path granted by routing module. We can observe the path from 33 to 49

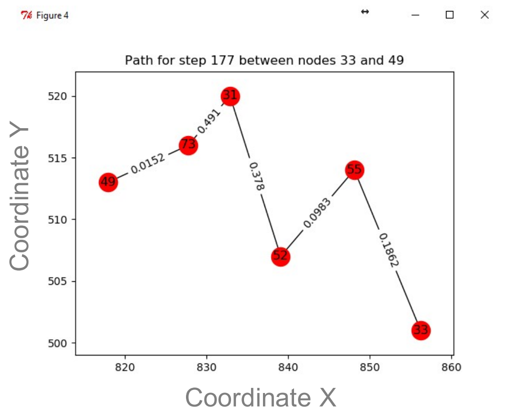


Figure 4.2.3b shows a path granted by propagation module. We can observe the path from 33 to 49

4.3 V2I and Cooperation Scheme

In the scenarios 3, 4 & 5, we have only modified the amount of RSUs to test I2V and V2I. We activated RANDOM ACCIDENT MODE and routing module to generate a complete simulation. The RSUs were placed with center in 466, 633, 800, 966, and 1133 on X. We could see a V2I message from RSU 0 (366-566) to car 30 shown in Figure 4.3.1a. Then, in Figure 4.3.1b is shown the V2I message. As we mentioned previously, V2I is limited to communicate in single mode.

Step 927 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ ~~~~~~~ [Enter] Car 30 entered RSU 0 at (372.93, 500.00) RSU 0 sent an emergency message to Car 30 with accidents [199] Car 30 will send an emergency message to Cars 10	Step 844 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ ~~~~~~~ Car 49 sent emergency message to RSU 0 via Car 49 -> RSU 0 with accidents [183,184
Figure 4.3.1a shows in yellow, the message displayed when an RSU inform a car.	Figure 4.3.1b shows in green, the message displayed when a car delivers a message on RSU.

In Figure 4.3.2, we can observe that a message eventually can get the RSU through V2V scheme by delivery in a single hop.

Car 48 will send an emergency message to Cars 72, 73, 52	Car 48 sent
emergency message to Car 72 via 48 -> 71 -> 52 -> 72 with accidents [9]	Car 48 sent
emergency message to Car 73 via 48 -> 71 -> 52 -> 72 -> 73 with accidents [9]	Car 48 sent
emergency message to Car 52 via 48 -> 71 -> 52 with accidents [9] Car 52 will send an emergency message to Cars 48, 71	Car 52 sent
emergency message to Car 48 via 52 -> 71 -> 48 with accidents [10, 9]	Car 52 sent
emergency message to Car 71 via 52 -> 71 with accidents [10, 9]	Car 71 sent
emergency message to RSU 0 via Car 71 -> RSU 0 with accidents [10,9]	

Figure 4.3.2 shows in orange, the message when a car delivers a message in RSU. The messages were coming car by car until getting the RSU.

On the other hand, comparing V2V in scenario 4 and 5 among the 3 types of routing: propagation, routing and cooperation criteria. Cooperation runs with 80-20% criteria for routing. Therefore, the result of propagation and the result of cooperation are likely to be similar. For this scenario, routing vs cooperation is certainly different unlike cooperation vs propagation. The following Figure 4.3.3 shows the differences in Single Accident Mode for routing vs Cooperation.

Car 22 sent emergency message to Car 32 via 22 -> 24 -> 35 -> 32 with accidents [0]
Car 22 sent emergency message to Car 32 via 22 -> 45 -> 24 -> 35 -> 32 with accidents [0]

Car 22 sent emergency message to Car 35 via 22 -> 24 -> 35 with accidents [0]
Car 22 sent emergency message to Car 35 via 22 -> 45 -> 24 -> 35 with accidents [0]

Car 22 sent emergency message to Car 41 via 22 -> 24 -> 35 -> 32 -> 41 with accidents [0]
Car 22 sent emergency message to Car 41 via 22 -> 45 -> 24 -> 35 -> 32 -> 41 with accidents [0]

Car 22 sent emergency message to Car 24 via 22 -> 24 with accidents [0]
Car 22 sent emergency message to Car 24 via 22 -> 45 -> 24 with accidents [0]

4.3.3 Shows a comparison of routing vs cooperation. In yellow, the result from routing module and in green, the result from cooperation criteria.

4.4 Scenario “Fewest”

For this scenario, we have the smallest number of cars. Simulation ran with 10 autos in 1000 sec. We could observe that even though simulation ran with very few autos, platform still have got ad hoc networks. There were moments in which no retransmission or any communication occurred as it was expected in a live “low density” network. We can see the behavior in Figure 4.4.1.

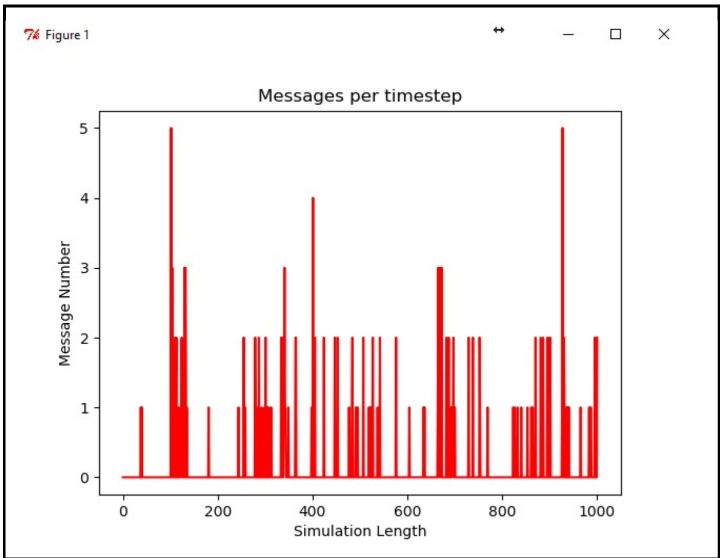


Figure 4.4.1 shows a chart with the result from “very few” autos, resulting in lack of communication in several moments in the whole simulation

In some time points the number of messages increased drastically because of nature of events as bottleneck. On the contrary, sometimes there are no messages to form a network because of lack of vehicles. Now, in Figure 4.4.2, we can observe a histogram, the coefficient of variance can grant us an idea how the population behaves through the time. In this scenario, some samples were suddenly raising thus the result was around 2.6. It means, there is heterogeneity within population.

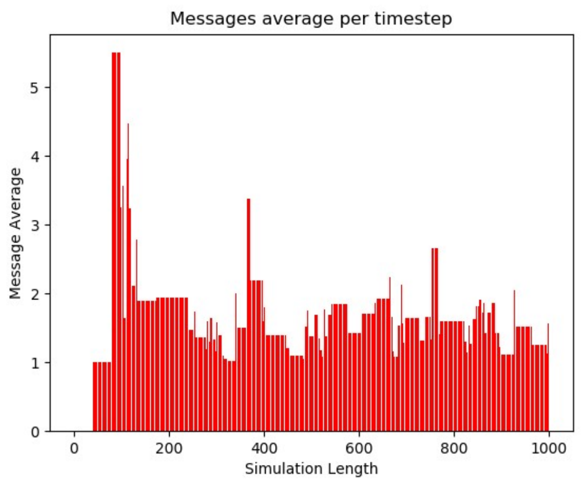


Figure 4.4.2. Average of sent messages along the time with variance = 0.514320843991 and mean=0.197802197802. The result is the coefficient of variation = 2.6

4.5 Scenario “Full”

Now, the mobility pattern changed to emulate “high density” traffic with full resources on play. We went on 2000 autos in 1000 seconds. There were 5 RSUs along the road. Once again, we ran RANDOM MODE to find the sample. We have selected a high density of retransmissions. In Figure 4.5.1 According to the expectations, it can be observed how retransmission increased in “high density” population.

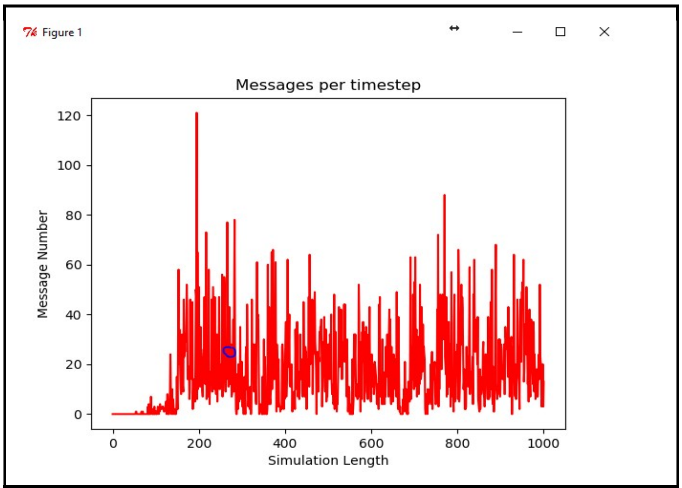


Figure 4.5.1 shows the result of a scenario with “high density” traffic.

This following chart Figure 4.5.2 is the histogram with variance and mean value. According to the coefficient of variance which was greater than 1, there were a lot of heterogeneity. Traffic event as jams increased undoubtedly messages exchange.

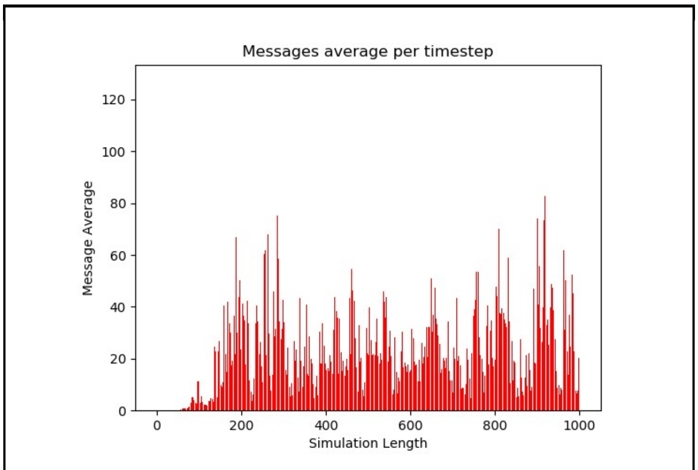


Figure 4.5.2 shows the average of sent messages along the time with variance = 511.586046321 and mean = 23.29304

In step 266, and for auto 190, we did a SINGLE ACCIDENT MODE. Figure 4.5.3a shows the isolated behavior for this sample. In Figure 4.5.3b the complete graph can be observed how the neighbors increased for the source node.

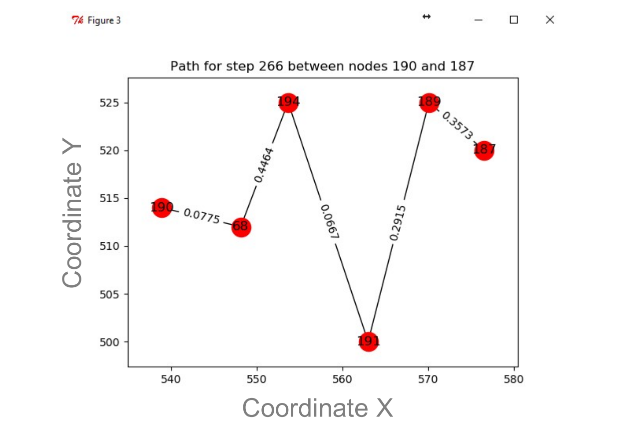


Figure 4.5.3a shows the view of isolated source and target

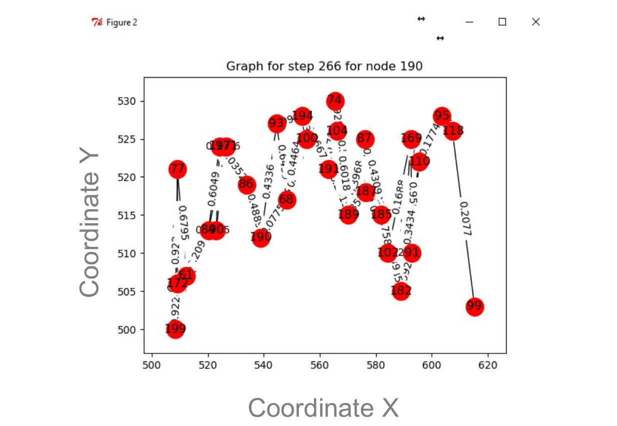


Figure 4.5.3b shows the view of all of neighbors of node 190

Finally, we summarized if the tool meet “networking challenges”. Definitely, our approach has demonstrated that was able to handle with communication schemes and routing mechanism. Figure 4.5.4 shows the results of challenges.

Networking Challenges			
Scenarios	V2V	I2V	V2I
1	☑		
2	☑		
3	☑	☑	☑
4	☑	☑	☑
5	☑	☑	☑
Fewest	☑	☑	☑
Full	☑	☑	☑

Table 4.5.4 shows the networking challenges. The tool was analyzed under V2V, V2I and V21 communication schemes.

Chapter 5

Conclusion and Future Works

As it has been seeing, our platform worked successfully with different types of communication: V2V, V2I. Among them, scheme V2V uses more computational time due to large groups formed by. Until up to 2000 nodes were simulated because of the space of the simulating area. This phenomenon got increased message exchange among vehicles. The tool proved a good support in scalability with all the variables active.

On the other hand, having emergency messages worked properly to test developed routing ability. Thus, we completed comparisons between routing modules to test their functionality. In this way, common protocols can be integrated later, or even new protocols could be tested. Undoubtedly, this capacity opens up a great opportunity to test different scenarios.

In addition to calculate the statistics as variance, mean value, it shows the results on charts to have a visual approach of behavior. By this mean, planning and sizing can be performed prior to designing networks. On the other hand, a summary file gets generated where contains source id, target id, number of hops.

In a next step, mobility trace could be modified dynamically if a bidirectional link to connect network to mobility be developed to exchange information. In the same way, our platform will work with different vehicular mobility pattern such a SUMO or STRAW. Events as safety or traffic messages could be modeled to optimize and influence transportation and traffic. Therefore, the networking approach gets impacted and could be evaluated simultaneously.

In a future stage, modeling data traffic, packet modeling, security will be factors to increase complexity in networking model. In the same way, propagation service with more complex process to increase realistic phenomena could be added. Variables as reflection, diffraction, and scattering which are involved in the channel could be modeled and included in the propagation service.

Not only for propagation but also for any service, a more complexity can be included. These models can be provided directly to the sourced in the location src>logic>routing>module meeting the contract previously mention. In addition, MAC and PHY layer can be included to improve propagation phenomena.

Contemporary platforms have not been implemented as parallel or distributed frameworks. However, there is still work to do in order to improve them in despite of they are sequential programs. To look for scalability improvement is the most important factor to aim, shortly.

Abbreviations and acronyms

VANETs Vehicular Ad-hoc Networks
V2V Vehicle-to-Vehicle Communications
V2I Infrastructure-to-Vehicle
RSU Road-Side Unit
ITS Intelligent Transportation Systems
OBU On-Board Unit
DSRC Dedicated Short Range Communications
TA Trusted authority
RVC Roadside-to-vehicle communication system
IDM-IM Intelligent Driving Model with Intersection Management
IDM-LC Intelligent Driving Model with Lane Change
IVC Inter-Vehicle Communications systems
RVC Roadside-to-Vehicle Communications
HVC Hybrid Vehicular Communication System
SIVC Single-Hop-Inter-Vehicle Communications
MIVC Multi-Hop-Inter-vehicle Communications
IDM Intelligent Driver Model

Bibliography

[1] M. L. Sichitiu and M. Kihl. Inter-vehicle communication systems: A survey. Communications Surveys & Tutorials, IEEE 10(2), pp. 88-105. 2008.0,

[2] F. F. C. B. Marco Fiore, Jerome Harri, “Vehicular mobility simulation for VANETs," Simulation Symposium, 2007. ANSS '07. 40th Annual, vol. 12, no. 4, pp. 301{309, 2007.

[3] H. Wu, R. Fujimoto, and G. Riley, “Analytical models for information propagation in vehicle-to-vehicle networks,” in Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th, vol. 6, 2004, pp.4548 – 4552.

[4] T. W. Chim, S. M. Yiu, L. C. K. Hui and V. O. K. Li, "VSPN: VANET-Based Secure and Privacy-Preserving Navigation," in *IEEE Transactions on Computers*, vol. 63, no. 2, pp. 510-524, Feb. 2014..

[5] S. T. Rakkesh, A. R. Weerasinghe and R. A. C. Ranasinghe, "A decentralized vehicle re-routing approach using vehicular ad-hoc networks," *2016 Sixteenth International Conference on Advances in ICT for Emerging Regions (ICTer)*, Negombo, 2016, pp. 201-207.

[6] W. Shen, L. Liu, X. Cao, Y. Hao and Y. Cheng, "Cooperative Message Authentication in Vehicular Cyber-Physical Systems," in *IEEE Transactions on Emerging Topics in Computing*, vol. 1, no. 1, pp. 84-97, June 2013.

[7] F. K. Karnadi, H. M. Zhi, and K. C. Lan, “Rapid generation of realistic mobility models for vanet,” in Wireless Communications and Networking Conference, 2007.WCNC 2007. IEEE, march 2007, pp. 2506 –2511.

[8] M. S. Ahmed, M. A. Hoque and P. Pfeiffer, "Comparative study of connected vehicle simulators," *SoutheastCon 2016*, Norfolk, VA, 2016, pp. 1-7.

[9] D. Auroux, L. Lin, H. Menouar, M. N. Mariyasagayam and M. Lenardi, "Integrated networking simulation environment for vehicular networks," *2008 IEEE Intelligent Vehicles Symposium*, Eindhoven, 2008, pp. 973-978.

,[10] S. Zemouri, S. Mehar and S. M. Senouci, "HINTS: A novel approach for realistic simulations of vehicular communications," *2012 Global Information Infrastructure and Networking Symposium (GIIS)*, Choroni, 2012, pp. 1-6.

[11] W. Arellano, I. Mahgoub and M. Ilyas, "Veins extensions to implement a message based algorithm for Dynamic Traffic Assignment in VANETs simulations," *2014 11th Annual High Capacity Optical Networks and Emerging/Enabling Technologies (Photonics for Energy)*, Charlotte, NC, 2014, pp. 29-35.

[12] C. Sommer and F. Dressler, "Progressing Towards Realistic Mobility Models in VANET Simulations," IEEE Comm. Magazine, vol. 46, no. 11, pp. 132-137, Nov. 2008.

[13] L. Wischhof, A. Ebner, H. Rohling, M. Lott and R. Halfmann, "Adaptive broadcast for travel and traffic information distribution based on inter-vehicle communication," *IEEE IV2003 Intelligent Vehicles Symposium. Proceedings (Cat. No.03TH8683)*, 2003, pp. 6-11.

[14] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in Proc. MobiCom, 1998, pp. 85–97.

[15] F. Bai, N. Sadagopan, and A. Helmy, "IMPORTANT: A Framework to Systematically Analyze the Impact of Mobility on Performance of Routing Protocols for Ad-hoc Networks", In Proc. of IEEE INFOCOM-2003, pp. 825-835, March-April 2003.

[16] F. J. Martinez, M. Fogue, M. Coll, J. C. Cano, C. T. Calafate and P. Manzoni, "Assessing the Impact of a Realistic Radio Propagation Model on VANET Scenarios Using Real Maps," *2010 Ninth IEEE International Symposium on Network Computing and Applications*, Cambridge, MA, 2010, pp. 132-139.

[17] L. Azpilicueta, C. Vargas-Rosales and F. Falcone, "Intelligent Vehicle Communication: Deterministic Propagation Prediction in Transportation Systems," in *IEEE Vehicular Technology Magazine*, vol. 11, no. 3, pp. 29-37, Sept. 2016.

[18] F. F. Jerome Harri and C. Bonnet, \Mobility models for vehicular ad hoc networks: A survey and taxonomy," IEEETrans. on Commun., vol. 11, pp. 1941, 2009.

[19] J. Härri, M. Fiore, F. Filali, and C. Bonnet, ”A Realistic Mobility Simulator for Vehicular Ad Hoc Networks”, Technical Report RR-05-150, Institut Eurecom, January 2007.

[20] VanetMobiSim – Vehicular Ad hoc Network mobility extension to the CanuMobiSim framework 2005-2006 Institut Eurécom/Politecnico di Torino