Journal of Communications Technology, Electronics and Computer Science, Issue 19, 2016
ISSN 2457-905X

Simulation of Information Dissemination using Flooding in VANET

Ravi Tomar*, Manish Prateek, Hanumat G.Sastry
School of Computer Science
University of Petroleum and Energy Studies
Dehradun, INDIA
Email: ravitomar7@gmail.com

***Abstract*— The rapid advancement in communication technologies, Intelligent Transportation System (ITS) has also gained its significance, aiming to improve efficiency and safety of transportation activities. Vehicle to Vehicle(V2V) and Vehicle to Infrastructure(V2I) are the main contributors of ITS architecture. The term used for communication among the vehicles is referred as Vehicular Ad-Hoc network(VANET), VANET is special kind of Mobile Ad-hoc Network(MANET), rapid change in mobile nodes(car/vehicles) in case of VANET makes it different from MANET, while sharing all other features of MANET. VANET has many advantages including minimizing traffic, reducing car accidents, co-operative awareness, environmental safety by monitoring CO2 emission. For successful utilization of this special network there is a need of method through which these nodes can communicate. The information flow needs to be maintained among all vehicles. The nature of this network is ad-hoc and highly mobile in nature, so routing is bit complex for enabling information exchange. Broadcasting comes as a solution to efficiently disseminate the information among nodes. However, there are various complexities involved in broadcasting. This paper explains and implements the simplest flooding-based broadcasting protocol. For developing protocols and algorithms to achieve above stated features out of VANET/ITS one need to test them through simulation. There are lots of Network simulators available which can simulate ad-hoc network but again coming to the complex part of VANET it has dense nodes with high mobility in VANET, obviously it is not possible through network simulators to simulate city traffic where each vehicle is considered as node, so a need to model the city scenario into nodes and edges structure evolves which can be further fed to network simulators to test algorithms efficiency and protocols correctness, in proposed work we will be modelling the scenario of Clock Tower, Dehradun city, the route contains two lane roads and other link roads along with buildings and parks. Modelling will be done using SUMO and Open StreetMap will be converted to SUMO networks and finally simulations will be done using VEINS through OMNET++.**

Keywords*— *VANET, Broadcasting, SUMO, OMNeT++, VEINS, ITS,OpenStreetMaps,VANET Modelling, VANET Simulation Introduction

I. INTRODUCTION

VANET is a special kind of Ad-hoc network with special features such as high mobility. For successful utilization of this special network there is a need of method through which these nodes can communicate. The information flow needs to be maintained among all vehicles. The nature of this network is ad-hoc and highly mobile in nature, so routing is bit complex for enabling information exchange. Broadcasting comes as a solution to efficiently disseminate the information among nodes. However, there are various complexities involved in broadcasting. Simulation is needed where physical implementation is not feasible to test some rough concepts, among many road traffic and safety is one of the major problem concern of this modern world.

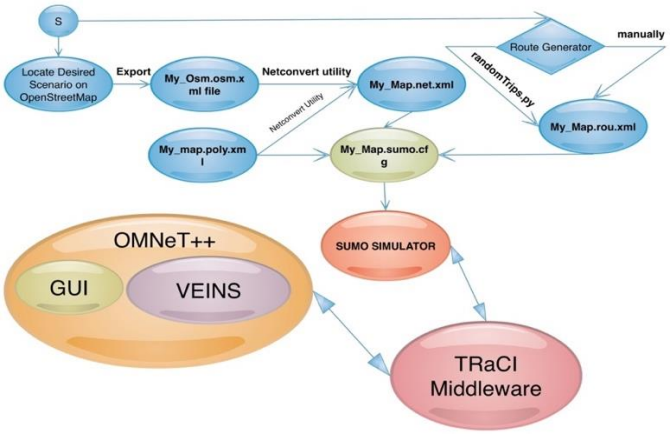

Figure 1 Process flow for entire simulation

Critical traffic problems such as accidents and traffic congestion require the development of new transportation systems[1].The need for some mechanism comes where one can model the real traffic in simulation and then dive in to propose solutions to the problems like traffic congestion, alternate route finding, minimizing red light stoppage, Co2 emission by controlling the flow of vehicles and many more. Post modelling it is needed to make those simulated vehicles intelligent so to make decisions and establish communication among each other for coming to some decision.VANET has a major challenge of Vehicles come and lose contact in a matter of seconds as speed is measured in miles per hour. [2] Here comes network simulators in picture which considers these vehicles as nodes and enables them to communicate with each other for real time decision making and behaving. During simulation data is processed and an overall decision is been

taken for platoons of vehicles through which they behave. To start with modelling we need to have some geographic location such as city or some specific area, which will be our area where we want to do some simulation. For this purpose we will be taking maps from OpenStreetMap[3][4],these maps will be then fed into SUMO[5] for modelling and then VEINS[8] framework will be used to make a realistic Vehicular Simulation using mobility from SUMO and Network from OMNeT++[7]. Any VANET application which needs simulation will obviously have to pass through all above mentioned phases. However there are different tools and techniques available to attain the goal each having its own merit and demerits . We present the mostly used and reliable method which also simulates as realistic as possible, without sacrificing speed and resources. The flow of entire process is shown in Figure 1.This paper is classified into 7 Sections, Section 2 elaborates OpenStreetMap and importing of maps from OpenStreetMaps, Section 3 explains about SUMO and modelling the map, Section 4 is dedicated to the OMNeT++ Simulator and Section 5 explains how VEINS framework is a comprehensive suite for SUMO and OMNeT++. Section 6 discuss the simulation parameters and results and finally Section 7 concludes the entire work done.

II. OPEN STREET MAP

OpenStreetMap (OSM) is a collaborative project by community and group of researchers to create a free editable map of the world.

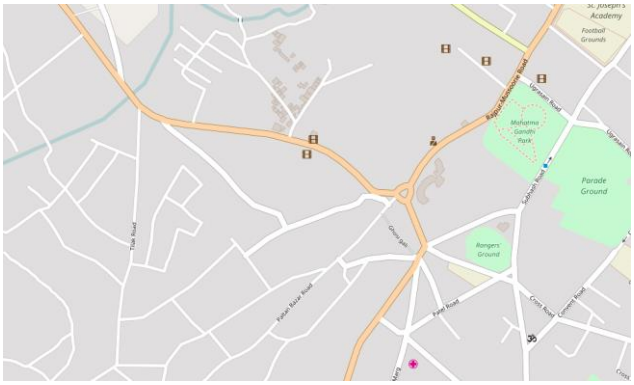

Figure 2 Map of Dehradun City

OpenStreetMap® [3]is *open data*, licensed under the Open Data Commons Open Database License (ODbL) by the OpenStreetMap Foundation (OSMF).User can simply visit the open street website[2] and export the specific geographic location, downloaded file would be a *.osm.xml file which is accepted by mostly all the simulators available. The downloaded osm file needs to be converted into *.net.xml file to be used by SUMO simulator. There are two ways of doing this the simpler one is the three clicks scenario generation. Python script is available inside tools folder of SUMO package, describe simply helps in downloading the map from open Street map and converting to SUMO supported format very comfortably and easily.Syntax for this method is :

python <***SUMO_HOME***>/tools/osmWebWizard.py

Figure 2 shows the selected area to be downloaded.The other method is using a utility named netconvert which is provided along with the sumo bundle. The net convert utility is used to convert the osm.xml file to the network file *.net.xml of

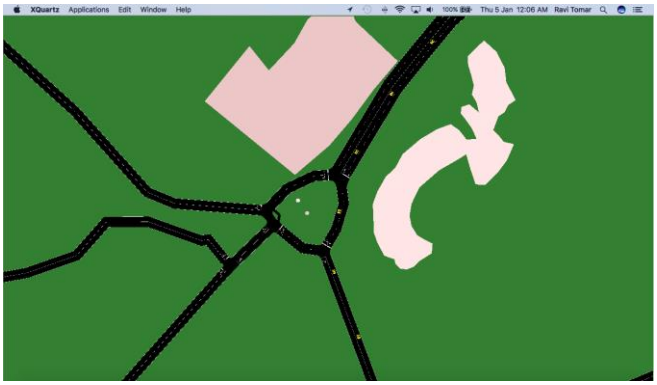

Figure 3 SUMO Simulation Model

SUMO. Syntax is given below :

netconvert --osm mymap.osm.xml -o mymap.net.xml.

The first method is easier to generate the first model but to generate custom models we need to do the manual method that is hundred convert which also takes different inputs during its execution like buildings hurdles left-hand drive etc. Further information can be seen in [4].

III. SUMO – SIMULATION OF URBAN MOBILITY

SUMO is a free and open traffic simulation suite which is available since 2001 and is used to model traffic scenario's**.** SUMO[4] provides GUI interface which can be started by typing sumo-gui at terminal once installation is done and path is set. SUMO comes with lots of tools available for achieving modelling rapidly, one such example is discussed in importing OpenStreetMaps, Other example is to create random traffic on road network which can be generated through randomTrips.py script available along with sumo bundle. SUMO creates a network by converting all roads to be the edges(*.edg.xml file) and all vehicles to be the nodes, as VANET is mobile network so all nodes moves along with the edges, and this movement is defined in route file(*.rou.xml), route file may contain individual path for all nodes or a flow can be created for platooning vehicles on same path at regular intervals. Apart from this sumo also facilitates by providing a polygon file (*.poly.xml) which is basically the buildings and parks etc. in real world along the edges. These three files acts as mandatory input to the SUMO simulator which are mentioned in a configuration file (*.sumo.cfg), this configuration file along with all three supporting files are given as input to sumo-gui simulator. We have created a model of city along with traffic by now , we can see the running nodes in Figure 3, yellow cars are nodes and black strips are road while rest polygons are buildings.

IV. OMNeT++

OMNeT++ is an open-source object-oriented modular discrete event network simulation framework. It has a generic architecture [7]. It is used to model computer networks or just as well for queueing network simulations. OMNeT++ provides only the necessary framework for developing a certain simulation module, and these models are developed independently in OMNeT++, they follow their own release cycles. This means that several frameworks can be modelled in the same research area [8]. OMNeT++ itself is not a concrete simulator but it is totally a building block kind of framework, it comprises of C++ library consisting Simulation kernel and some basic utility classes such as random number generator, topology discovery etc. It just provides an infrastructure to create simulations and these configurations are done in *.ned files. Figure 4 shows the abstract working scenario of any OMNeT++ Simulation. We start with collecting the modules available which will be used for our scenario and all modules will communicate through message exchange among themselves. Then a NED file is created which models the hierarchy and structure of all modules, all the active modules needs to be coded in C++ so the working logic can be implemented there and finally define omnetpp.ini file which is the final point of passing values to the simulation. All these are compiled and linked to the OMNeT Simulator kernel which gives us the simulation based on our scenario generation, as output finally OMNeT creates two kind of files output vector and output scalar files. These files can be used to analyze and validate the process using any available tool like R, visualization tool is also provided in the OMNeT IDE.

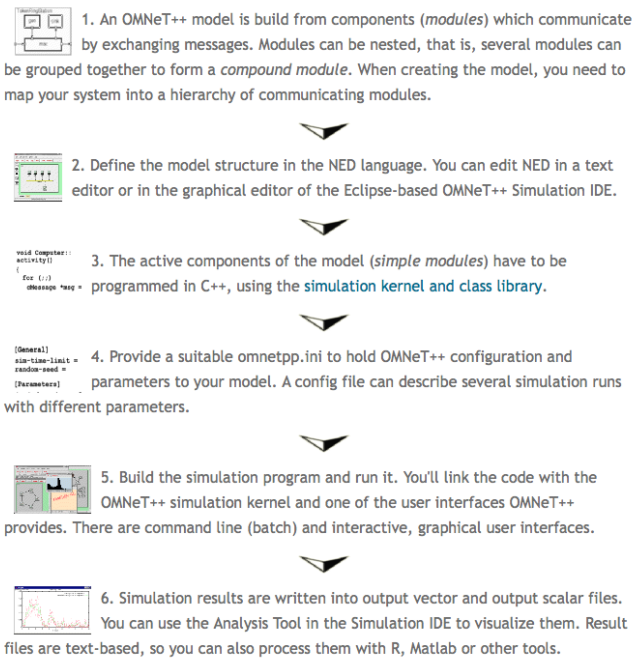

Figure 4 Working Sequence for OMNET++ Simulation[10]

V. VEINS (VEHICLES IN NETWORK SIMULATION)

Veins is an open source framework for running vehicular network simulations. It is based on two well-established simulators: OMNeT++, an event-based network simulator, and SUMO, a road traffic simulator[11]. Models generated in veins are executed by an event-based network simulator (OMNeT++) while interacting with a road traffic simulator (SUMO). Other components of Veins take care of setting up, running, and monitoring the simulation. Veins provide a all at one place suit of IVC-specific modules, Figure 5 shows the VEINS framework, we can see all modules are running under OMNeT++ Kernel and simulation control is maintained by OMNeT++ while Mobility is provided by SUMO through TRaCI server. VEINS come with many modules and are used as per our requirement, it also contains various obstacles modules which depicts real world building interference during simulation. VEINS finally gives us output in 2 files of OMNET which are text based scalar and vector output, which can be further analyzed to get results.

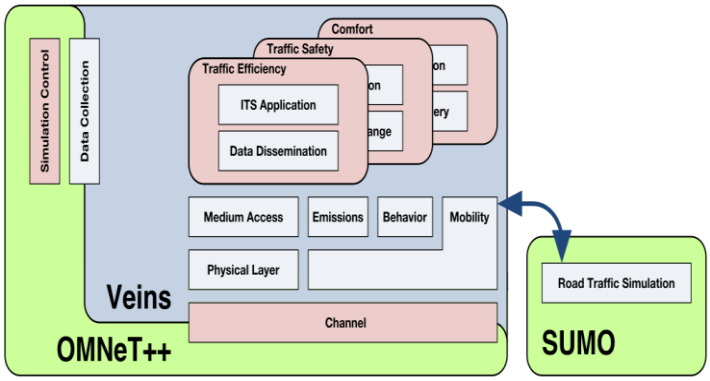

Figure 5 VEINS Framework

VI. SIMULATION

Now, we have understood all the tools and framework used to model and simulate any scenario, here we have done the simulations through VEINS framework on OMNET++ which is connected to SUMO via local client server model, and to establish this we take help of TraCI (TRaffic Control Interface) which is distributed along with SUMO bundle in src folder.[8] TraCI works as middleware to connect SUMO and OMNET++.VEINS framework which is running inside

OMNET++ provide a python script named *sumo-launchd.py*, this script creates a client-server connection between SUMO TraCl server connection and VEINS via OMNET++, Figure 6 shows the sever waiting on terminal listening on port 9999 at localhost. When simulation starts on OMNET++ it sends a request to TRaCI server and gets nodes location in response, thereby creating a realtime road network on OMNET++ Simulation panel. This step is to share SUMO mobility on

OMNET++, now comes the role of Network Simulator OMNET++ . which implies communication between the nodes and make vehicles exchange data. Using Veins framework in OMNET++ we simulate our scenario and various parameters are specified in omnetpp.ini file which is required by OMNeT simulator to run simulations. Specific parameters such as playground size which is taken as 25000m by 25000m and simulation time is set to 6000s. Configuration parameters for WaveAppLayer used for communication between nodes/vehicles, here applType represents the protocol used for information exchange. 802.11p specific parameters

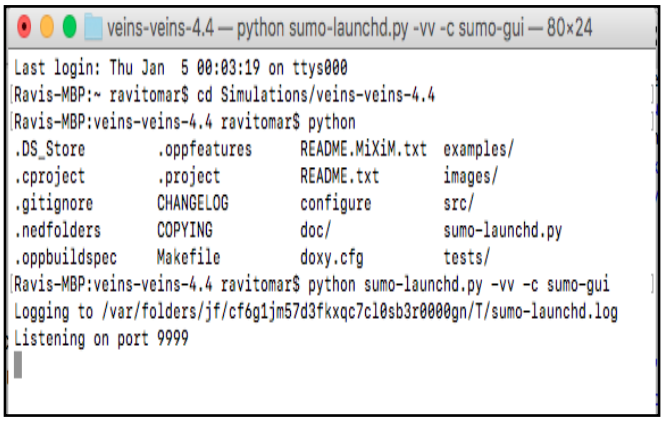

Figure 6 VEINS TRaCI middleware script running

for Network Interface cards like Transmission Power, carier frequency, bitRate etc. TraCI specific settings, here we can see that we are sending our request to localhost and on port number 9999 where SUMO TraCI server is already started and Mobility parameters, from here we can configure number of accidents which can happen to any node, at what time accident is scheduled and by when accident will be resolved.

VII. CONCLUSION

This paper demonstrated the modelling and simulation of real world scenario/traffic and scheduled an accident, which acts as an event generator for all following vehicles, vehicles receiving signals again rebroadcasts the event information to other vehicles in vicinity. The algorithm automatically changes the route of following vehicles once accident is detected and threshold time is lapsed. Figure 7 shows Node 110-Node 116 are been rerouted towards destination when Node 50 is unable to recover from accident for so long.

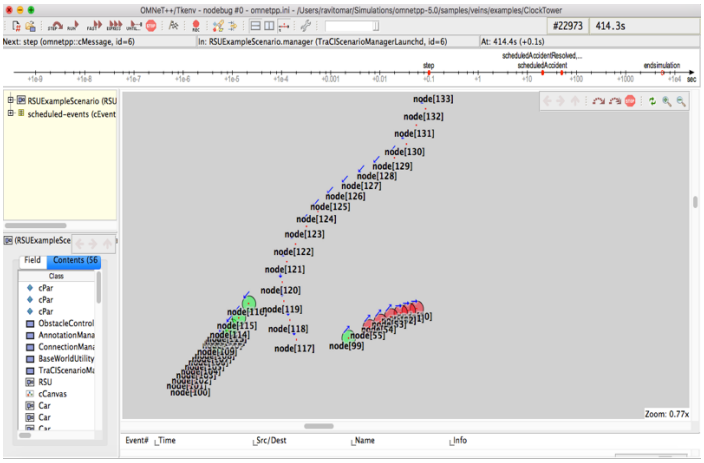

Figure 7 Simulation of Vehicles on OMNET++(RED one waited for more than 10sec) (Left One are re-routed because of accident)

REFERENCES

[1] *V. Srivastava, M. Motani*, Cross-layer design: a survey and the road ahead, IEEE Communications Magazine 43 (12) (2005) 112–119.

[2] *R Tomar, M Prateek, GH Sastry*, Vehicular Ad-Hoc Network(VANET)-An Introduction, IJCTA,International Science Press 9(18) 2016,pp. 8883-8888.

[3] *Wikipedia* (2016, October 04). Retrieved from https://en.wikipedia.org/wiki/OpenStreetMap

[4] *Wikipedia* (2016, October 04). Retrieved from http://sumo.dlr.de/wiki/Networks/Import/OpenStreetMap

[5] *DLR and contributors, SUMO Homepage* (2016, October 04).Retrieved from http://www.dlr.de/ts/desktopdefault.aspx/tabid-9883/16931_read-41000/

[6] *Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, and Laura Bieker*,"Recent Development and Applications of SUMO – Simulation of Urban MObility", International Journal on Advances in Systems and Measurements, vol 5 no 3 & 4, year 2012 128-137.

[7] András Varga and team,OMNeT Homepage(2016,October 04).Retreived from https://omnetpp.org/

[8] *Christoph Sommer, Reinhard German and Falko Dressler*, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," *IEEE Transactions on Mobile Computing*, vol. 10 (1), pp. 3-15, January 2011.

[9] *Wikipedia*(2017,January 04).Retreived from http://www.sumo.dlr.de/wiki/TraCI

[10] *OpenSim Ltd, OMNET++ Webpage* (2016, October 04).Retrieved from https://omnetpp.org/documentation/3632

[11] *Christoph Sommer and Team,VEINS Homepage(*2016,October 04) Retreived from http://veins.car2x.org