Simulating Vehicle-to-Vehicle D2D communication in OMNeT++ with SUMO and SimuLTE

[1] A.K Rohit, [2] Mamatha T, [3] Aakanksha Ashok, [4] Akhil R Kothakota, [5] B C Siddhu Siddharth
[1] Software Engineer, [2] Assistant Professor, RVCE, [3]Student,RVCE, , [4]Student,RVCE, , [5]Student,RVCE,
[2] Department of Computer Science & Engineering,
[1] Nokia, Bangalore, India

Abstract : ***Vehicle to Vehicle communication is a fast growing technological field in which vehicles are able to communicate with each other wirelessly; information about speed, position, jams, pollutions, delays or accidents on road. But VANET communication faces challenges like low bandwidth, short transmission range, link failure due to high mobility of vehicles during message transmission. D2D communication enables offloading of traffic from core-network and establishing communication between devices. The resulting advantage of such a system is increase in spectral efficiency, reduced latency. The main focus of this paper is to provide information on the working of each simulator and how to configure these simulators to work in unison to provide a vehicle-vehicle D2D simulation with a network model and a real world traffic model between two vehicles at the intersection. In this paper D2D communication between vehicles using SUMO and SimuLTE integrated with the OMNeT++ framework is simulated. These simulation configurations are presented so that they can be recreated to study V2V in higher detail.***

IndexTerms - D2D, V2V, SUMO, SimuLTE, OMNet++, VeinsLTE

I. Introduction

In recent years, traffic related accidents have been growing at an alarming rate, especially at intersections [1]. This growth can be attributed to a number of reasons - low visibility, blind spots, faulty traffic lights etc. Since these accidents are expected to only increase in the future, a new solution, apart from the existing traffic technology is needed. Advances in the wireless communication domain has made the passing of critical information between vehicles feasible. One such advancement is the application of D2D to Vehicle-to-Vehicle communication.

Vehicle-to-Vehicle or V2X (Vehicle-to-Anything), is the ability of a vehicle to wirelessly transmit information about its speed, position, direction and other safety critical parameters to nearby vehicles [2]. Thereby enabling the driver to “see” or be warned of potential hazards that other sensor based warning systems cannot detect. This could lead to huge reductions in the number of road accidents occurring everyday [4]. V2V could also potentially ease congestion and reduce emissions related pollution.

A communication system which is reliable, robust and has low latency even in high density situations is needed. We are simulating the V2V communication at intersections and junctions because it is here that the application of V2V is most obvious. Intersections have the highest densities of vehicles, and as mentioned before are highly prone to accidents for a variety of reasons.

There are existing standards for V2X comm. Such as Direct Short Range Communication (DSRC), based on IEEE 802.11p. It operates on 75 MHz of bandwidth in the 5.9 GHz region. There is ongoing research in the field of Cellular-V2X which includes LTE-V2X, d2d-V2X and the upcoming 5G-V2X. Cellular V2X is an improvement over DSRC/802.11p which is not suitable for active safety use cases[5].

D2D is a recent standard in LTE communication. It is the direct communication between two mobile units (UEs or User Equipment) without an intermediate Base station or Core network. Here instead of an uplink and downlink mechanism, transmission occurs through sidelink between UEs. It allows for some of the communication data to be offloaded from the eNodeBs and hence increases the throughput. This is particularly of importance in high density situations like intersections where a large number of UEs are present within a single cell. This direct communication between nearby vehicles will improve the energy efficiency and spectrum utilization [6].

II Related Work

In our simulation we integrate 3 simulators, SUMO, OMNeT++ and SimuLTE. Together they have been released as VeinsLTE.

OMNeT++ is a framework used for building network simulators. It is developed by OpenSim Ltd and written in C++. It is just a framework and does not have models for popular protocols like TCP/IP and hence external frameworks such as INET need to be integrated. INET provides models for simulating protocols at all layers of the internet stack, both wireless and wired communication.

SUMO (Simulation of Urban Mobility) is an open source road traffic simulator. It has many features that contribute to a very realistic traffic model.

SimuLTE is a tool for simulating LTE and LTE-A networks included in 3GPP Release 8 and beyond. D2D was introduced in 3GPP release 12. SimuLTE provides models for various LTE entities such as UEs, EnodeBs, etc

III THE SIMULATION

1. Creating a traffic scenario in SUMO.

SUMO is microscopic simulation of traffic, where each object in the traffic can be modelled [8]. This can be used to program each component such as vehicle, signals, etc. To simulate traffic we would need nodes(vehicles), paths(roads) and paths the nodes would take. SUMO(Simulation of Urban Mobility) simulator makes use of 3 basic files written in XML. These files are as follows:

Nodes
Edges

Routes

The node files would include the location of the nodes(represented on the XY plane) and an ID, which would be used as a reference in the other files.

The edge file is used to bridge two nodes together to form a virtual version of a road. This could be done by specifying the two node IDs, stated in the node XML file, being taken into consideration while creating a road. Each edge or route is given a specific ID which is referred to in other files.

The route file is used to give a static direction or a path as to where the virtual entities or cars are supposed to travel upon. This is done by specifying which route ID a particular vehicle is assigned to.

Once the required basic files are defines,the configuration file can be created by typing "sumo-filename.cfg"

2. Instantiating the vehicles in SUMO as UEs in OMNeT++

All entities in OMNeT++ are modelled as modules, which can be simple or compound. Modules communicate with other modules with the help of gates. This interface can pass messages such as packets or frames. Parameters such as propagation delay, data rate and error rates can be set to make the simulation more realistic. OMNet++ is object oriented and modules can be inherited. The topology and composition of each module is described using NED files. They contain the composing modules, gates and can be used to create an entire network. Required protocols are imported from INET and SimuLTE.

Another important file is the .ini initialisation file. Which is used to set some crucial parameters at the time of running. The parameters are discussed later on.

A UE is a complex module, consisting of many SUB modules. SimuLTE defines UEs and EnodeBs as shown in Figure.1 and Figure.2

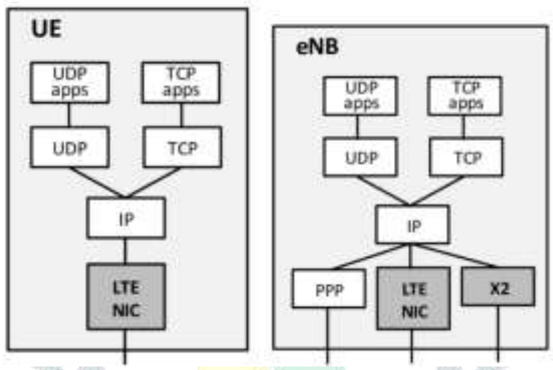

Figure.1 SimuLTE architecture[7].

INET provides the Network Interface Card for communication between an EnodeB and a UE for uplink and downlink in traditional LTE and LTE-Advanced architecture. This LTE NIC implements the LTE protocol stack. SimuLTE provides a D2D NIC for communication via sidelink between two UEs. EnodeBs communicate with each other via the X2 interface. A network of communicating EnodeBs forms a core network.

A vehicle should be instantiated as a "car" in OMNeT++. A car is a type of UE, but with additional modules to incorporate its mobility and D2D capability. The car.ned file as described in VeinsLTE is an extension of a UE with a Veins Mobility module. We modify car.ned to include a D2D capable Network Interface Card as shown in Figure.3

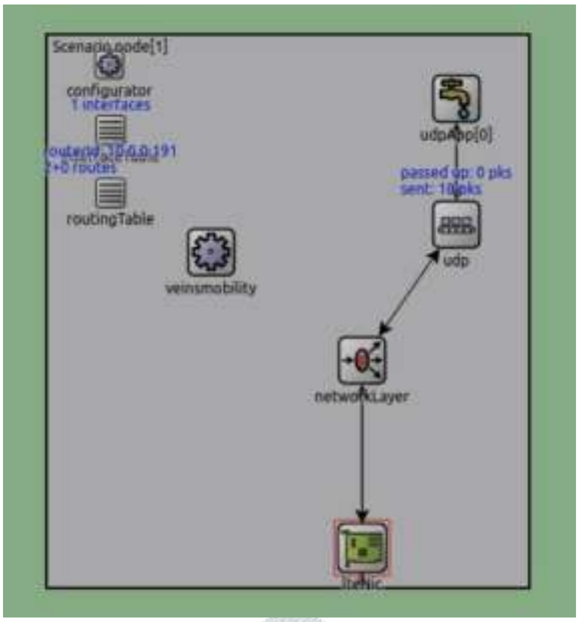

Figure.2 TRACI Scenario Manager

The TRACI Scenario Manager connects to a running instance of the SUMO launch script. It specifies how and when to make the connection to the TRACI server. It tells TRACI to create a car module (as described by the car.ned file for each vehicle in SUMO.

The TRACIScenarioManagerLaunchd.ned can my modified for this purpose as follows:

String moduleType = default("org.car2x.veins.nodes.Car")
String moduleName = default("node")

It also specifies what each instantiated vehicle will be called in Omnet++. We are calling each car as a node.

3. Creating a Scenario

A scenario is a scenario.ned file describing a module that contains the various elements in your simulation. These elements can be UEs, eNodeBs, routers, servers, buildings etc. Each of these elements is also a module and is a submodule of the scenario.

4. Setting up D2D communication between 2 Cars

To enable D2D communication between the cars, the nodes must have parameter d2dCapable set to true. This can be set in the .ini file. In the one-to-one case we need to configure the sender as a D2D sender and receiver as a D2D recipient.

*.eNodeB.d2dCapable = true
*.node[*].d2dCapable = true
**.amcMode = "D2D"

Other quality parameters can be set. CQI which is the Channel Quality Indicator carries information regarding how good/bad the communication channel is. UEs typically transmit CQI information to the network. The CQI reporting parameter can be set to True or False. Otherwise, the CQI can be defined as shown. The transmission and reception power can also be set. Other parameters like eNodeB height, antenna gain, loss, noise, thermal noise can be set in a separate xml file - configchannel.xml

*.eNodeB.nic.phy.enableD2DCqiReporting = false **.usePreconfiguredTxParams = true
**.d2dCqi = 7

The simulation aims to provide a sidelink of information transfer between the two communicating vehicles as opposed to the traditional uplink and downlink transfer between the enodeB and the respective communicating UEs. The sidelink transfer of information helps in decreasing latency by reducing the functionality of the enodeB to a mere resource allocation process.

6. Establishing the connection between SUMO and OMNeT++

SUMO needs to be invoked every time the OMNeT++ simulation is started up. The script to launch SUMO is located in sumo-launchd.py script. This script creates a proxy TCP connection between SUMO and OMNeT. The script will create a socket connection on port 9999 and wait for the simulation to start.

IV. RESULTS

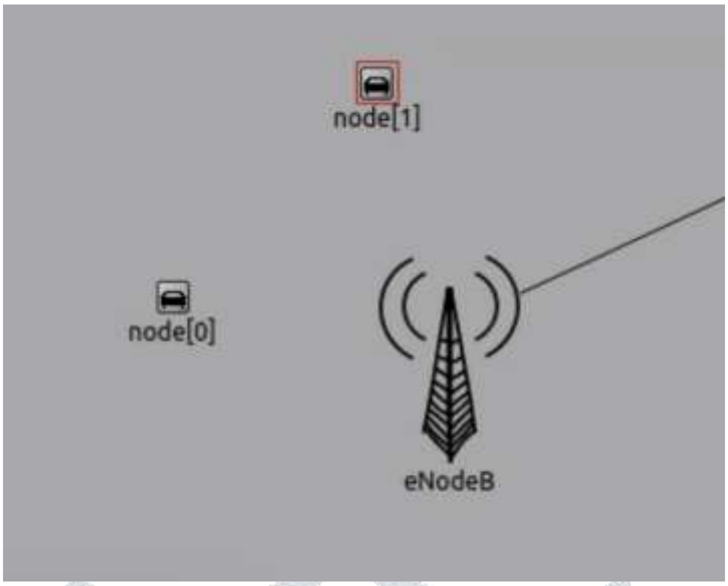

Figure.3 Snapshot of the OMNet++ point of view.

Figure.3 shows Snapshot of the OMNet++ point of view when the simulation is run.It shows the 2 nodes (node[0] and node[1]) communicating while acquiring resources for the connection from the eNodeB. Looking at the logs allows us to discern which packets are being transferred between the nodes, such as, Airframe packets(packets that the user defined) between the nodes and the HardReq packets (packets used to allocate resources) between the nodes and the eNodeB.

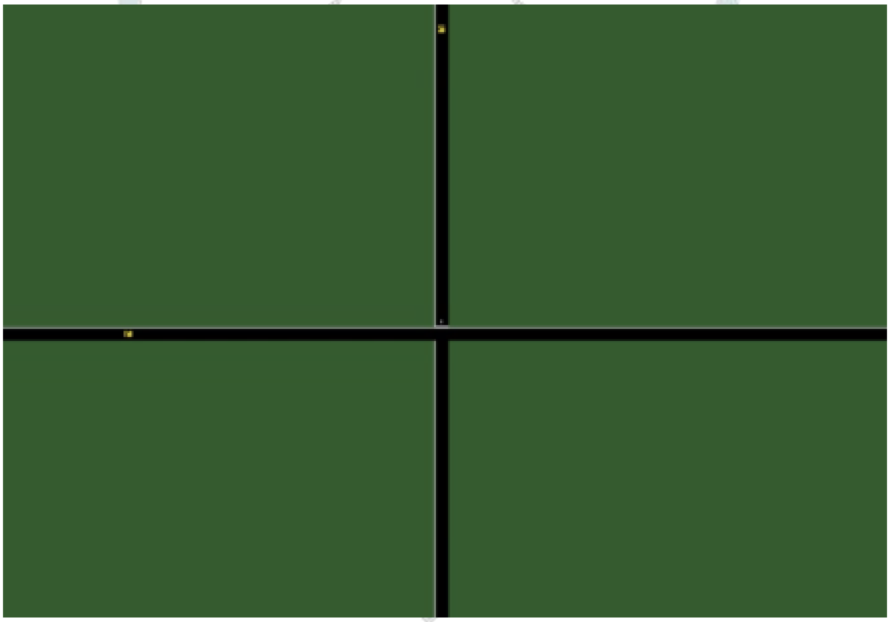

Figure.4 Snapshot of the simulation's SUMO point of view.

Figure 4 is the snapshot of the simulation from SUMO point of view.

This provides us with a real-world view of how the traffic would be in real-time.The yellow nodes represent the cars approaching the intersection. SUMO works in unison with OMNeT++ to obtain network data and hence simulate the position of the nodes accordingly. [8]

Figure.5 Close-up view of how nodes are rendered in SUMO's point of view of the simulation.

V CONCLUSION

In this paper the steps taken to simulate vehicle to vehicle communication between two vehicles at an intersection was presented.The steps taken to configure the SimuLTE and OMNet++ frameworks so that they can be integrated with the vehicular Simulator SUMO. This setup can be exploited to study the benefits and potential disadvantages of employing D2D communication between vehicles. Not just for safety critical applications but for any advanced use-cases in the future.

REFERENCES

[1].X. Yang, L. Liu, N. H. Vaidya and F. Zhao, "A vehicle-to-vehicle communication protocol for cooperative collision warning," The First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004., 2004, pp. 114-123.

[2].https://www.extremetech.com/extreme/176093-v2v-what-are-vehicle-to-vehicle-communications-and-how-does-it-work

[3].https://www.nhtsa.gov/technology-innovation/vehicle-vehicle-communication

[4].Harding, John, et al. Vehicle-to-vehicle communications: Readiness of V2V technology for application. No. DOT HS 812 014. 2014.

[5].Qualcomm, Leading the world to 5G: Cellular Vehicle-to-Everything (C-V2X) technologies, June 2016

[6].E.SREE HARSHA1, T. TIRUPAL2 “LTE-Advanced Cellular Networks for D2D Communications” International Journal of Scientific Engineering and Technology Research Volume.03, IssueNo.18, August-2014.

[7].Nardini, Virdis, Stea “Simulating device-to-device communications in OMNeT++ with SimuLTE: scenarios and configurations”, September 2016

[8].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, 5 (3&4):128-138, December 2012.