Rapid Generation of Realistic Mobility Models for VANET

Feliz Kristianto Karnadi* Zhi Hai Mo*
*School of Computer Science and Engineering
University of New South Wales
Sydney, NSW 2052, Australia
{fkka085,zhmo487}@cse.unsw.edu.au

Kun-chan Lan†
†National ICT Australia Ltd
Bay 15, Australian Technology Park
Eveleigh NSW 1430, Australia
Kun-chan.Lan@nicta.com.au

***Abstract*— One emerging, new type of ad-hoc network is the Vehicular Ad-Hoc Network (VANET), in which vehicles constitute the mobile nodes in the network. Due to the prohibitive cost of deploying and implementing such a system in real world, most research in VANET relies on simulations for evaluation. A key component for VANET simulations is a realistic vehicular mobility model that ensures conclusions drawn from simulation experiments will carry through to real deployments. In this work, we introduce a tool MOVE that allows users to rapidly generate realistic mobility models for VANET simulations. MOVE is built on top of an open source micro-traffic simulator SUMO. The output of MOVE is a realistic mobility model and can be immediately used by popular network simulators such as ns-2 and qualnet. We evaluate and compare ad-hoc routing performance for vehicular nodes using MOVE to that using the random waypoint model. We show that the simulation results obtained when nodes moving according to a realistic mobility model is significantly different from that of the commonly used random waypoint model.**

***Index Terms*— VANET, Mobility, Simulation**

I. Introduction

Vehicular Ad-Hoc Network (VANET) communication has recently become an increasingly popular research topic in the area of wireless networking as well as the automotive industries. The goal of VANET research is to develop a vehicular communication system to enable quick and cost-efficient distribution of data for the benefit of passengers' safety and comfort.

While it is crucial to test and evaluate protocol implementations in a real world environment, simulations are still commonly used as a first step in the protocol development for VANET research. Several communication networking simulation tools already exist to provide a platform to test and evaluate network protocols, such ns-2 [1], OPNET [2] and Qualnet [3]. However, these tool are designed to provide generic simulation scenarios without being particularly tailored for applications in the transportation environment. On the other hand, in the transportation arena, simulations have also played an important role. A variety of simulation tools such as PARAMICS [4], CORSIM [5] and VISSIM [6] etc have been developed to analyze transportation scenarios at the micro- and macro-scale levels. However, there was little effort in integrating communication techniques and scenarios in a realistic transportation simulation environment.

One of the most important parameters in simulating ad-hoc networks is the node mobility. It is important to use a realistic mobility model so that results from the simulation correctly reflect the real-world performance of a VANET. A realistic mobility model should consist of a realistic topological map which reflect different densities of roads and different categories of streets with various speed limits. Another important parameter should be modeled is the obstacles. In the real world, a vehicle node is typically constrained to streets which are separated by buildings, trees or other objects. Such obstructions often increase the average distance between nodes as compared to an open-field environment. In addition, each vehicle needs to decide a turning directions at the intersection (e.g. turn left, turn right or go straight). Such a turning model could have an effect on the congestion of the road as well as the clustering of the vehicles. Furthermore, a smooth deceleration and acceleration model should be considered since vehicles do not abruptly break and move. Many prior studies [7], [8] have shown that a realistic mobility model with sufficient level of details is critical for accurate network simulation results.

The time-consuming process from collecting traces, analyzing the data to finally generating and implementing models can make some research obsolete before it is finished. In this work, we develop a tool MOVE (MObility model generator for VEhicular networks) to facilitate users to rapidly generate realistic mobility models for VANET simulations. Our tool MOVE is built on top of an open source micro-traffic simulator SUMO [9]. The output of MOVE is a mobility trace file that contains information of realistic vehicle movements which can be immediately used by popular simulation tools such as ns-2 or qualnet. MOVE allows users to rapidly generate realistic VANET mobility models in two aspects:- by interfacing with real world map databases such as TIGER [10] and Google Earth, MOVE allows user to conveniently incorporate realistic road maps into the simulation. In addition, by providing a set of Graphical User Interfaces that automate the simulation script generation, MOVE allows the user to quickly generate realistic simulation scenarios without the hassle of writing simulation scripts as well as learning about the internal details of the simulator. The architecture of MOVE is shown in shown in Figure 1.

The rest of this paper is structured as follows. In Section II we describe the architecture and implementation of MOVE. We evaluate MOVE by comparing the mobility models generated by MOVE to the commonly used Random Waypoint Model in Section III. We provide related work in Section IV. Finally, conclusions and future work are presented in Section V.

II. ARCHITECTURE

MOVE is currently implemented in Java and runs atop an open-source micro-traffic simulator SUMO. MOVE consists of two main components: the Map Editor and the Vehicle Movement Editor, as shown in Figure 2. The Map Editor is used to create the road topology. Currently our implementation provides three different ways to create the road map – the map can be manually created by the user, generated automatically, or imported from existing real world maps such as publicly available TIGER (Topologically Integrated GEographic Encoding and Referencing) database from U.S. Census Bureau [10]. The Vehicle Movement Editor allows the user to specify the trips of vehicles and the route that each vehicle will take for one particular trip. We currently support three different methods to define the vehicle movements – the vehicle movement patterns can be manually created by the user, generated automatically, or specified based on a bus time table to simulate the movements of public transportations. The information users input in the Map Editor and the Vehicle Movement Editor is then fed into SUMO to generate a mobility trace which which can be immediately used by a simulation tool such as ns-2 or qualnet to simulate realistic vehicle movements. Users can also visualize the generated mobility trace by clicking on the "Visualization" button on the main menu, as shown in Figure 3.

One of the major overhead before one can start conducting research using simulations is to learn about the internal details of the simulator and write customized simulator-specific scripts to generate various simulation scenarios for the research problem under study [11]. To reduce such an overhead, MOVE also provides an interface to automatically generate simulation scripts on the fly based on the parameters that the user inputs into MOVE. We currently support auto-generation of ns-2 and qualnet simulation scripts, as shown in Figure 4.

A. MAP Editor

In MOVE, the road map can be generated manually, automatically or imported from a real world map. Manual generation of the map requires inputs of two types of information, nodes and edges. A "node" is one particular point on the map which can be either a junction or the dead end of the roads. Furthermore, the junction nodes can be either normal road junctions or traffic lights. The edge is the road that connects two points (nodes) on a map. The attributes associated with an edge include speed limit, number of lanes, the road priority and the road length. Figure 5 shows snapshots of nodes editor and edge editor.

We have also integrated Google Earth into MOVE to facilitate the creation of nodes in a realistic setting. Google

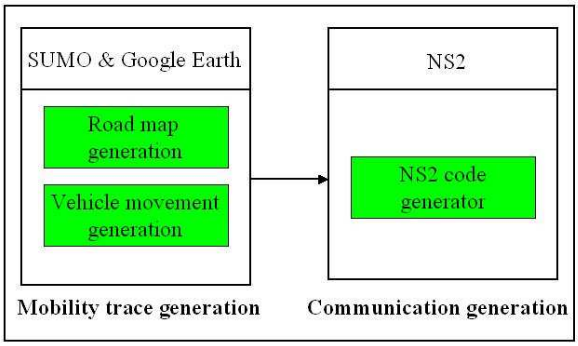


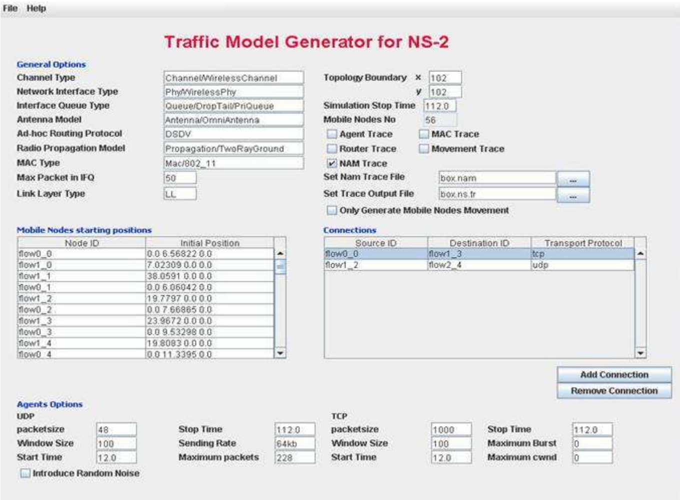
Fig. 1. The architecture of MOVE

Earth is a tool that enables its user to view the satellite image map of any place on earth. One of the functionality that Google Earth provides is called "placemark" which allows the user to put a mark on any location of the Google Earth map. Each placemark contains the longitude and latitude information of the selected locations and can be saved into a file in KML format [12]. Hence, one can define the *node* location on the Google map and then extract the node information by processing the saved KML file. This allows MOVE users to generate a map for any real-world road on earth for their simulations. Figure 6 shows an example of using Google Earth to generate nodes for the major intersections in the Eastern Suburb of Sydney.

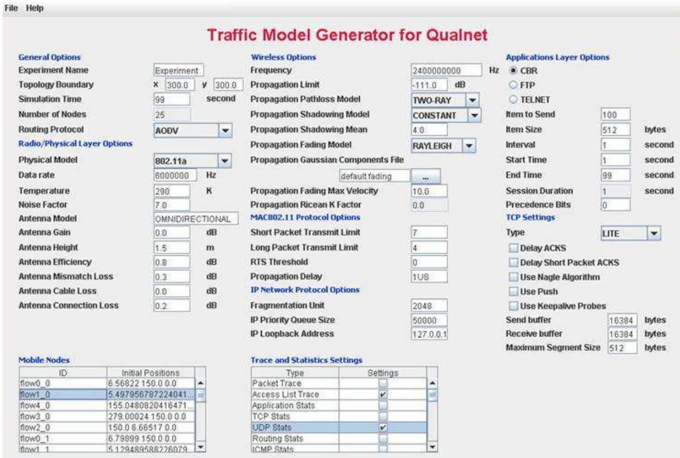
The road map can also be generated automatically without any user input. Three types of random maps are currently available: grid, spider and random networks. There are some parameters associated with different types of random maps such as number of grids and the number of spider arms and circles. Finally, one can also generate a realistic map by importing real world maps from publicly available database. We currently support the TIGER maps which are available from U.S. Census Bureau. Figure 7 shows a grid map generated from the random map generator and a street map in the Houston area based on a TIGER database file.

B. Vehicle Movement Editor

The movements of vehicles can be generated automatically or manually using the Vehicle Movement Editor. To generate vehicle movement automatically, one needs to first define a vehicle *flow* which describes a fleet of vehicles toward the same direction. The parameters of each *flow* consist of the starting road and destination of the flow, the time to start and end the flow, the number of vehicles in the flow and the inter-departure time of the vehicle originating from the starting road. In addition, a MOVE user can define the probability of turning to different directions at each junction (e.g. 0.5 to turn left, 0.3

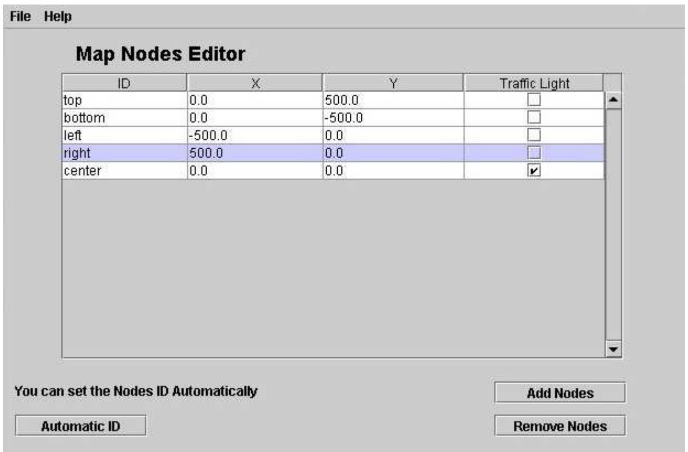


(a) Traffic model generation for ns-2

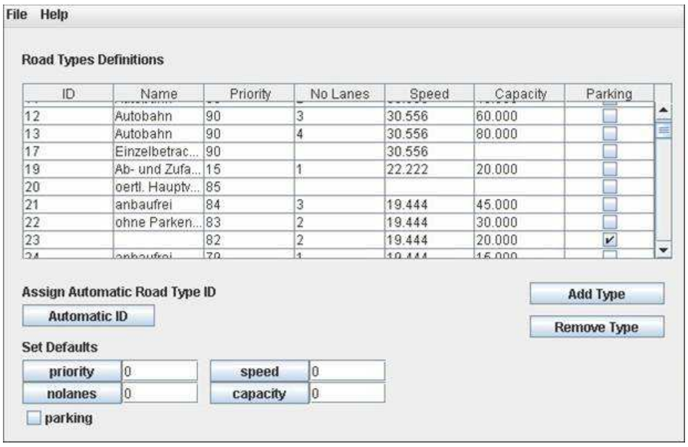


(b) Traffic model generation for qualnet

Fig. 4. Traffic Model Generation



(a) Node Editor



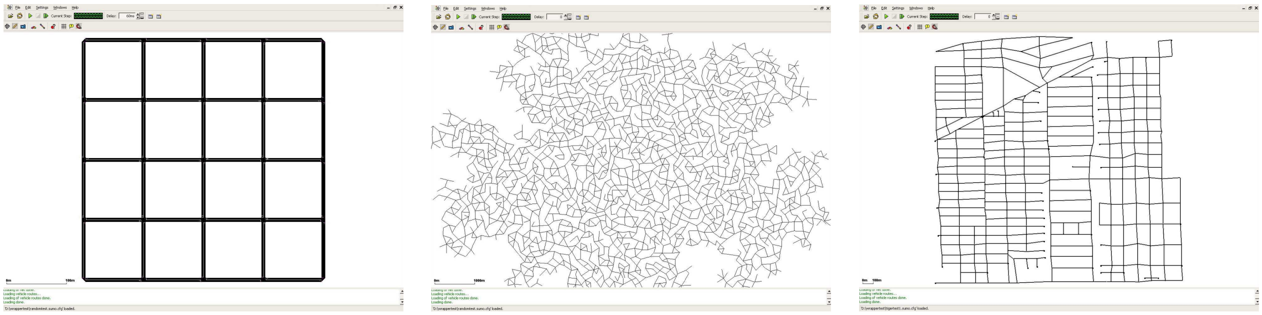
(b) Edge Editor

Fig. 5. Road Map Generation

to turn right and 0.2 to go straight) in the editor. Figure 8(a) shows a snapshot of the Flow definition Editor.

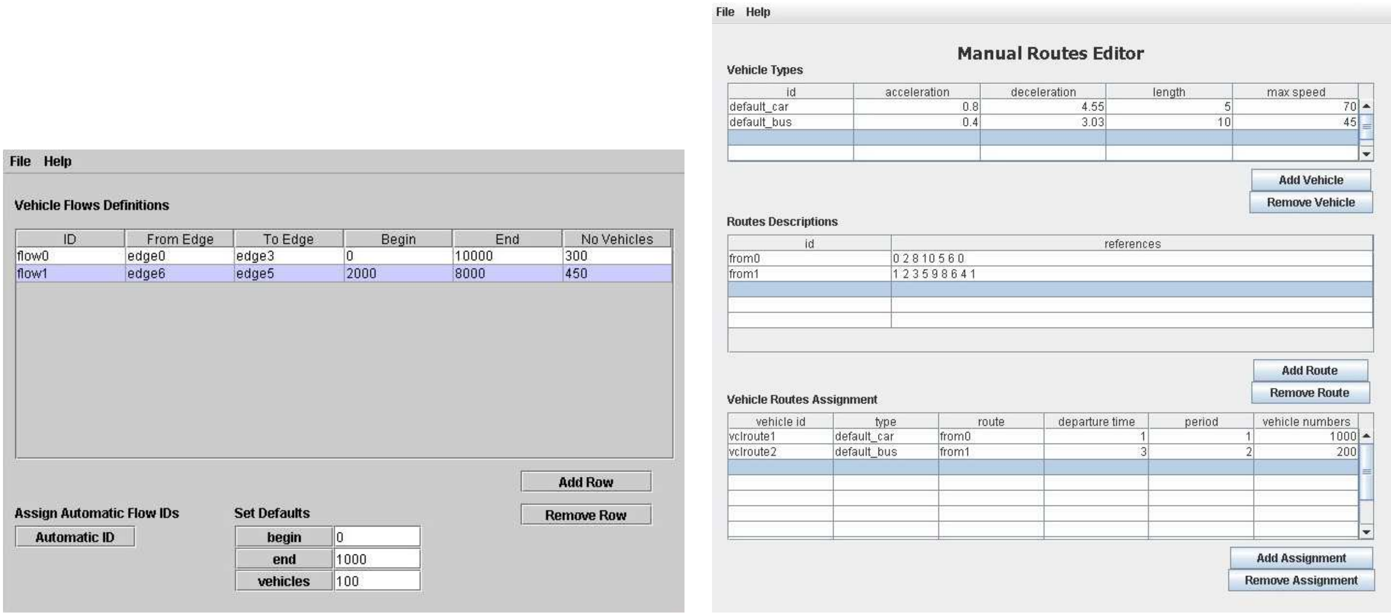
One can also generate vehicle movement manually using the Vehicle Movement Editor which allows users to specify several properties of vehicle routes including the number of vehicles in a particular route, vehicle departure time, origin and destination of the vehicle, duration of the trip, vehicle speed (including acceleration, deceleration and maximum speed), etc. Figure 8(b) shows a snapshot of the Vehicle Movement Editor. Note that, in addition to simulating vehicle-to-vehicle communication, our tool is also useful for simulations of vehicle-to-infrastructure (e.g. the communication between mobile nodes and road-side static gateway nodes). A static node can be created in MOVE by assigning the vehicle with a maximum speed of zero in the Vehicle Movement Editor.

On-board communication has recently become an increasingly popular research topic. A new paradigm of *Networks in Motion* is quickly attracting interest from the research community and is also being viewed as a viable commercial solution for extending Internet services to public transport passengers. MOVE allows users to enter the bus time table to simulate the movements of public transport. We model the bus as one type of vehicle which has similar parameters, such as speeds, routes, etc, associated with it as other vehicles. In addition, one needs to define the departure times of the first and the last bus and the bus inter-arrival time (which is assumed to be constant in our implementation) to simulate the bus time table. Figure 9 shows the editor for entering the bus



(a) A grid map (b) A random map (c) A real world map

Fig. 7. Road Map generation using the Map Editor



(a) Vehicle Flow Generation (b) Vehicle Movement Editor

Fig. 8. Vehicle Movement Generation

route information.

III. Evaluation

In this section, we evaluate the impact of mobility models generated by MOVE on the performance of ad-hoc routing protocol. We compare the performance of AODV [13] when used with the random waypoint model to that using the MOVE mobility model.

The simulation experiments were carried out in ns-2. Each simulation lasts for 900 seconds. We generated scenarios for 150 nodes moving in an area of 4 square kilometres. We varied the number of source nodes from 10 to 50, each of which is a CBR traffic source transmitting UDP packets of a size 64 bytes at the rate of 4 packets per second. All nodes use 802.11 MAC operating at 2Mbps. The transmission range is 250m. The propagation model employed in the simulation is the log normal shadowing model. We used a path loss exponent 2.56 with standard deviation 4.0 based on real-world measurement data from an inter-vehicle experiment we previously carried out in Sydney suburban area. The road topology generated by MOVE is based on the TIGER database data.

Figure 10 shows the packet delivery ratio of AODV with different number of traffic sources. Each data point represents the average of six runs and the error bars represent the range of observed packet delivery ratios. Overall, the packet delivery ratios increase as the number of traffic sources increases, which suggest a higher density of nodes can increase the network performance as long as the increasing density does not create more radio interference. In addition, the packet delivery ratios of AODV when using MOVE mobility models

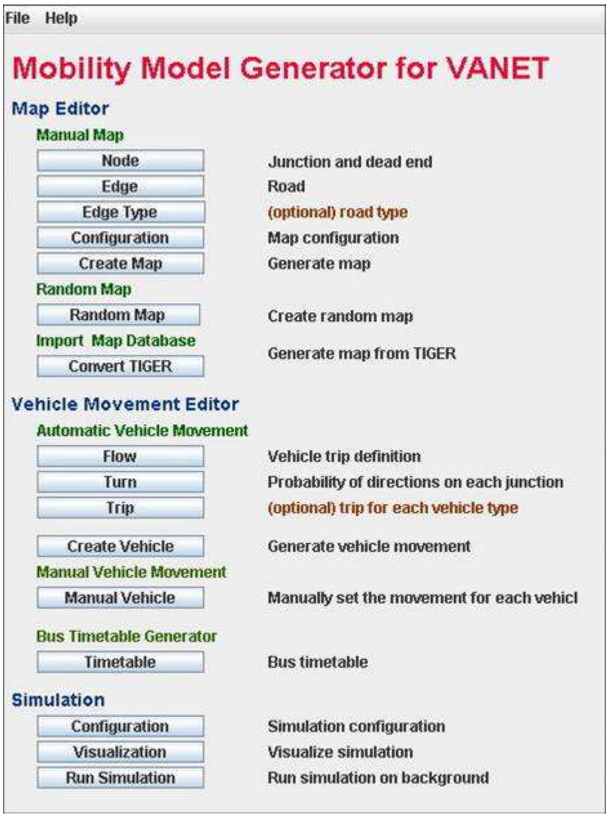


Fig. 2. Mobility Model Generator

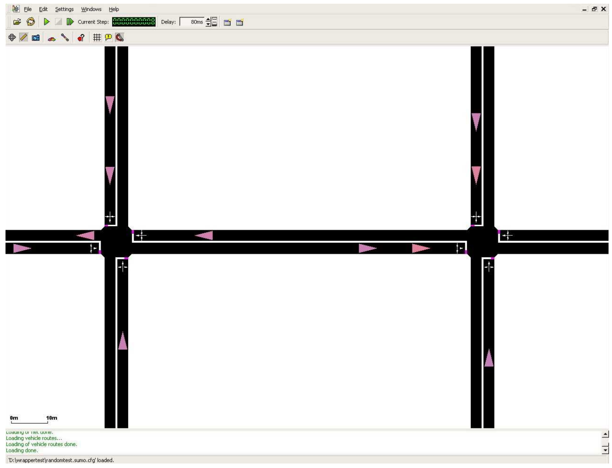


Fig. 3. Visualization of Vehicle movements

Fig. 6. Generating realistic map using Google Earth

are lower than when using Random Waypoint model and have larger variations. The larger variance in MOVE data points is possibly due to unstable network connectivity imposed by constrained node movements by roads and traffic control mechanisms (such as traffic lights). Figure 10 clearly shows that the simulation results using a more realistic mobility model can be drastically different from that using a simplistic open field model. Note that our results are also consistent with prior work [14].

IV. Related Work

Groovesim [15] is a topography-accurate street-map based vehicle network simulator and is based on GrooveNet, a geographic routing protocol for vehicular networks. It provides several different modes of operation. In Drive Mode, GrooveSim can process data from a GPS unit to provide a real-time map of the vehicle's current location. It can also be used as an emulator in Hybrid Simulation Mode where real vehicles on the road and virtual vehicles in the simulation can interact with each other. Groovesim also provides a tool for analyzing the simulation results. One limitation of Groovesim is that it is strongly tied to one specific routing protocol (i.e. GrooveNet), which limits its use for simulating other routing protocols in a VANET environment. In addition, GrooveSim does not provide mobility traces for network simulators.

STRAW [14] is an extension of SWANS (Scalable Wireless Ad Hoc Network Simulator) [16], a Java-based simulator for wireless simulations. STRAW contains simulation tools for generating mobility models and traffic models and is also able to use real street maps like TIGER data to build the road topology. However, currently the mobility models can be supported by STRAW is limited. For example, while STRAW supports multiple lanes, the vehicles are not allowed to change lane and the starting position is not configurable. Another drawback of this tool is its dependency on SWANS. Finally, STRAW does not provide any GUI that allows the users to visualize the movements of cars.

BonnMotion [17] is a simple tool that can be used to create and analyses mobility scenarios. Similar to MOVE, the mobility scenarios created by BonnMotion can be exported to ns-2 and qualnet. However, BonnMotion only models basic motion constraints and does not consider any micro-mobility. Furthermore, BonnMotion is a text-based application that runs on a command shell and does not provide any graphical user

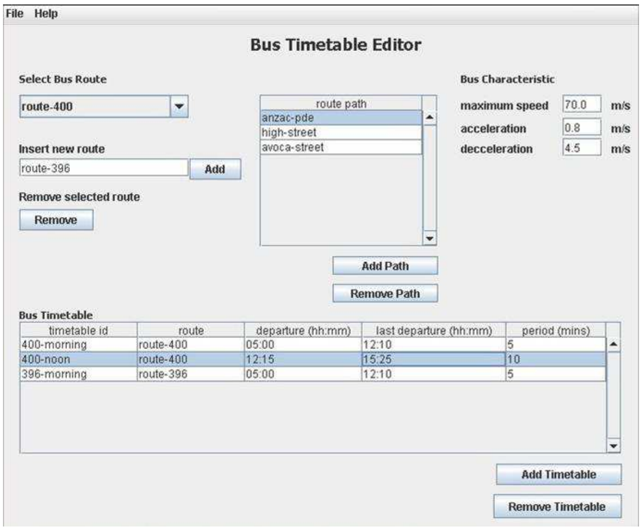


Fig. 9. Bus route generation

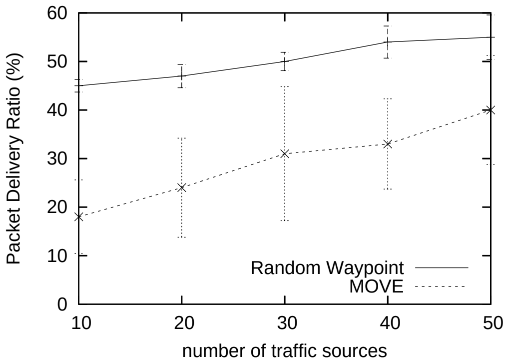


Fig. 10. Comparison between Random Waypoint and model generated by MOVE

interfaces as MOVE does.

Complementary to these previous efforts, our work emphasizes on creating a tool that allows users to rapidly generate realistic mobility models for VANET simulations.

V. Conclusion and Future work

In this paper, we describe our initial efforts in implementing a tool MOVE which is based on an open source micro-traffic simulator SUMO. MOVE allows user to quickly generate realistic mobility models for vehicular network simulations. In addition, MOVE provides an interface to automatically generate simulation scripts for ns-2 and qualnet. Finally, we show that the simulation results using MOVE is significantly different from that using the commonly used random waypoint model.

We have made MOVE publicly available and can be downloaded via the following URL - http://www.cse.unsw.edu.au/ klan/move/. Our next step is to use this tool to understand the effect of level of details (such as traffic light, stop sign, speed limit, road congestion, etc) in the context of VANET simulation. In our current implementation, the movements of vehicles are based on static configurations defined in the Vehicle Movement Editor. In other words, the mobility model is first generated off-line and then used by a network simulator like ns-2. In the next version of our software, we plan to build an interface to tightly integrate SUMO and ns-2. Such an interface will allow that vehicle state information (such as location, speed, direction, etc) can be fed into ns-2 in real time. Hence, during the simulation the vehicles can dynamically adjust their routes based on different traffic scenarios and communication techniques employed.

References

[1] The Network Simulator ns 2. http://www.isi.edu/nsnam/ns/index.html.

[2] OPNET Simulator. http://www.opnet.com/.

[3] QualNet Network Simulator. http://www.scalable-networks.com/.

[4] Pamramics: Microscopic Traffic Simulation. http://www.paramics-online.com/.

[5] CORSIM. http://www.fhwa-tsis.com/.

[6] PTV simulation VISSIM. http://www.english.ptv.de/.

[7] Amit Kumar Saha and David B. Johnson. Modeling mobility for vehicular ad hoc networks. In *Proc. of the 2nd ACM International Workshop on Vehicular Ad Hoc Networks (VANET)*, October 2004.

[8] John Heidemann, Nirupama Bulusu, Jeremy Elson, Chalermek Intanagonwiwat, Kun chan Lan, Ya Xu, Wei Ye, Deborah Estrin, and Ramesh Govindan. Effects of detail in wireless network simulation. In *Proc. of Communication Networks and Distributed Systems Modeling and Simulation Conference*, January 2001.

[9] SUMO Simulation of Urban MObility. http://sumo.sourceforge.net/.

[10] TIGER (Topologically Integrated GEographic Encoding and Referencing). http://www.census.gov/geo/www/tiger/.

[11] Sally Floyd and Vern Paxson. Difficulties in simulating the Internet. *ACM/IEEE Transactions on Networking*, 9(4):392–403, February 2001.

[12] KML tutorial. http://www.keyhole.com/kml/kml_tut.html.

[13] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc on demand distance vector (aodv) routing, July 2003. RFC 3561.

[14] David R. Choffnes and Fabián E. Bustamante. An integrated mobility and traffic model for vehicular wireless networks. In *Proc. of the 2nd ACM International Workshop on Vehicular Ad Hoc Networks (VANET)*, Cologne, Germany, September 2005.

[15] R. Mangharam, D. S. Weller, D. D. Stancil, R. Rajkumar, and J. S. Parikh. GrooveSim: A topography-accurate simulator for geographic routing in vehicular network. In *Proc. of the 2nd ACM International Workshop on Vehicular Ad Hoc Networks (VANET)*, Cologne, Germany, September 2005.

[16] Zygmunt J. Haas and Rimon Barr. Density-independent, scalable search in ad hoc networks. In *Proceedings of IEEE International Symposium on Personal Indoor and Mobile Radio Communications*, September 2005.

[17] BonnMotion A mobility scenario generation and analysis tool. http://web.informatik.uni-bonn.de/iv/mitarbeiter/dewaal/bonnmotion/.