ISSN No. 0976-5697

Volume 4, No. 6, May 2013 (Special Issue)

International Journal of Advanced Research in Computer Science

REVIEW ARTICAL

Available Online at www.ijarcs.info

Vehicular Mobility Simulation For VANET

Miss.Shruti S.Khope
B.E-I.T (Final Year), Jawaharlal Darda Institute of Engineering and Technology,
Yavatmal. (MS)INDIA
shrutiskhope@gmail.com

Miss. Swati V.Gayakwad
B.E-I.T (Final Year), Jawaharlal Darda Institute of Engineering and Technology,
Yavatmal. (MS)INDIA
Gayakwad.swati43@gmail.com

Prof.Sandip T. Dhagdi
Assistant Professor, Jawaharlal Darda Institute of Engineering and Technology,
Yavatmal. (MS)INDIA
sandip.yml@gmail.com

Abstract: The study of vehicular ad-hoc networks (VANETs) requires efficient and accurate simulation tools. As the mobility of vehicles and driver behaviour can be affected by network messages, these tools must include a vehicle mobility model integrated with a quality network simulator. We present the first implementation of a well known vehicle mobility model to NS-3.Vehicular ad-hoc networks are self organised networks built up from moving vehicles, and are part of the broader class of Mobile ad-hoc networks (MANET).One of the main challenges posed by VANET simulation is the faithful characterization of vehicular mobility at both macroscopic and microscopic levels, leading to realistic non-uniform distributions of cars and velocity, and unique connectivity dynamics. In this paper, we first present and describe VanetMobiSim, a freely available generator of realistic vehicular movement traces for networks simulators.

Keywords: VanetMobiSim, Mobility model, lanechange model, IDM, MOBIL.

I. INTRODUCTION

Vehicular ad-hoc networks (VANETs) are networks in which each node is a vehicle. Such systems aim to provide communications between individual vehicles and between vehicles and nearby fixed equipment, or roadside units. The goal of VANETs, and more broadly vehicular networks, is to improve traffic safety by providing timely information to drivers and concerned authorities.VANET simulations have typically been segregated into traffic simulations and network simulations. Traffic simulators, such as CORSIM, SUMO, VISSIM, and VanetMobiSim have been used to generate realistic mobility traces of vehicle traffic. These traces would then be fed into well-known network simulators such as ns-2, QualNet, OPNET, or GloMoSim to measure network performance.

The problem with integrated simulators is that often either the mobility model is overly simplified or the network model is overly simplified. In order to study important networking properties of VANETs, a high quality network simulator is essential. We have chosen to balance these two concerns by taking the latest version of the highly-regarded network simulator, ns-3, and adding a well known traffic mobility model in order to provide an integrated simulator for VANET research. VANETs are distributed, self organizing communication networks built up from traveling vehicles, and are thus characterized by very high speed and limited degrees of freedom in nodes movement patterns. A critical aspect in a simulation study of VANETs is the need for a mobility model which reflects, as close as possible, the real behavior of vehicular traffic. When dealing with vehicular mobility modeling, we distinguish between macro-mobility and micro-mobility descriptions.

The mobility model proposed in this paper combines both trace and survey based advantages, by respectively denying routes destinations by exploiting geographical information from open-source maps and by using real trace counting data to denying and assess the validity of the generated vehicles mobility[1].

II. LITERATURE REVIEW

In Congested Traffic States in Empirical Observations and Microscopic Simulations [2], this paper has examined that to what extent the phase diagram can serve as a general description of collective traffic dynamics in open, inhomogeneous systems. The original phase diagram was formulated for on-ramps and resulted from simulations with macroscopic models. The proposed intelligent-driver model (IDM) is simple, has only a few intuitive parameters with realistic values, reproduces a realistic collective dynamics, and also leads to a plausible "microscopic" acceleration and deceleration behaviour of single drivers.

The paper Minimize Overall Braking decelerations Induced by Lane changes (MOBIL) [4] investigates the possible virtue of the modification of longitudinal and lane-change behaviours of drivers by intelligent cruise control systems that augment individual driver behaviour by enforcing minimum separation between vehicles. Here the driver behavioural model was modified to the behaviour models used, the IDM for longitudinal motion and MOBIL lane-changes were very good at re-creating observed traffic. Once tested, the models were then used to test the effects of various driver behaviours on traffic flows.

III. PROPOSED WORK

A. *Architecture:*

Here we describe the components of our design, which consists of five main classes (Figure 1):

a. *Vehicle* - a mobile node that contains a wireless communications device.
b. *Obstacle* - a *Vehicle* that has no mobility.
c. *Model* - the IDM car-following mobility model.
d. *LaneChange* – the MOBIL lane change model.
e. *Highway* - holds *Vehicle* and *Obstacle* objects and use a *Vehicle*'s *Model* and *LaneChange* properties to control its mobility.

Highway uses the first four classes to generate the traffic in a highway. Since vehicular mobility models, and especially car-following models like the one we implement, need to know the position and mobility of other vehicles, the *Highway* object must be used to control the mobility.

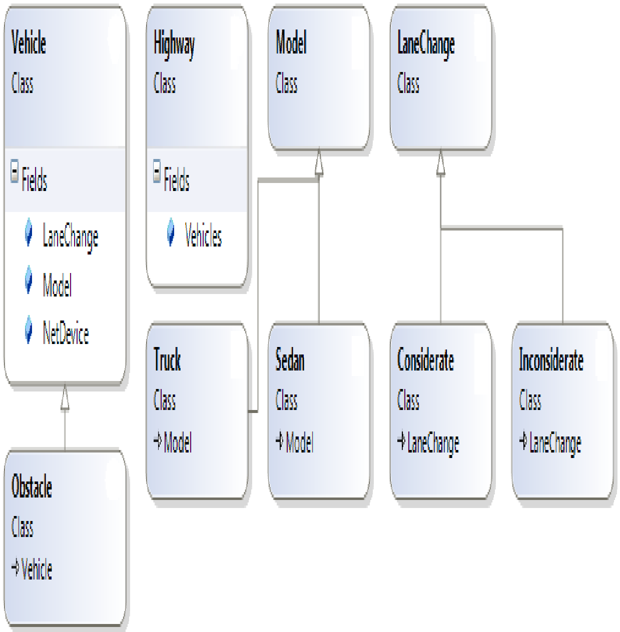

Figure 1: Class diagram of the main components in our design.

a. *Vehicle:*

A *Vehicle* is a mobile node that contains a wireless communications device. A *Vehicle* has the following properties:

a) VehicleID.
b) Width – width of the vehicle in meters.
c) Length – length of the vehicle in meters.
d) Lane – lane number on the highway where the vehicle is located.
e) Direction – {-1, 1} (Assume eastbound is 1 and westbound is -1).
f) Position – a vector (x, y, z), where x is the rear position of the vehicle, y is the center of the vehicle, and z is the altitude of the vehicle above the highway (all units in meters).
g) Velocity– in m/s.
h) Acceleration – in m/s^2.
i) Model – mobility model settings, desired velocity is associated with the mobility model.
j) Lanechange – lane change model settings.

In our design, the *Highway* object is in charge of managing the positions, directions, and the lane numbers of its vehicles. A *Vehicle*'s acceleration and velocity can be set manually or can be calculated based on the IDM mobility model rules. A *Vehicle* is able to change lanes, if necessary and if possible, based on the MOBIL lane change model.

b. *Obstacle*:

An *Obstacle* is a static node that contains a wireless communications device. It is inherited from the *Vehicle* class and has all of the capabilities of a *Vehicle* except that it cannot be mobile (i.e., velocity = acceleration = model = lanechange = 0). An obstacle can be used as a barrier to close a lane or to temporarily create stoppages that result in congestion on the highway. An obstacle can also be used as a roadside unit or other piece of infrastructure along, but outside of, the highway. If an *Obstacle* is placed on the highway, it must have a direction and lane number. Anything that can be done to a *Vehicle* object can be done to an *Obstacle* object (aside from affecting mobility), so in the rest of this paper we will just use the term *Vehicle*.

c. *Mobility Model:*

Model is the class that implements the mobility model for a *Vehicle*. We have implemented the Intelligent Driver Model (IDM) in ns-3 based on equations and parameters developed by Treiber [2][3]. IDM is a car-following model, meaning that each vehicle's acceleration or deceleration depends upon its own velocity, its desired velocity, and the position and velocity of the vehicle immediately in front in the same lane, which Treiber calls the *front vehicle*.

IDM uses parameters and the current state of the vehicle and front vehicle to compute the new acceleration. Acceleration is, in turn, used to update the velocity and position of the vehicle.

The function *CalculateAcceleration* in the *Model* class uses the IDM equations to calculate and return the new acceleration at each time step. The vehicle's new velocity and position are then adjusted based on this new acceleration.

When a vehicle is heading towards a traffic light intersection, it is informed by the macroscopic description about the state of the semaphore. If the color is green, passage is granted and the car maintains its current speed through the intersection. If the color is red, crossing is denied and the car is forced to decelerate and stop at the road junction by using the modified IDM parameters similarly to a stop sign.

d. *Lane Change Model:*

LaneChange is the class that implements the lane changing model for a *Vehicle*. We have implemented the MOBIL lane change model based on equations and parameters developed by Treiber [2] [3]. Each lane change in this model must satisfy both the safety criterion and the incentive criterion. The safety criterion states that the lane change must not cause the vehicle that is being changed in front of (the *back vehicle*) to decelerate unsafely (faster than a certain threshold) [4].

To allow for some variability in how aggressive drivers are in deciding when to change lanes. MOBIL weights the other vehicles' disadvantage with a politeness factor, *p*. When $p \geq 1$, the driver is considerate and puts others' disadvantages equal to or ahead of their own advantage. In reality, most drivers are in the $0 < p \leq 0.5$ range, where some weight is given to other drivers' disadvantage. If p=0, the driver is inconsiderate, discounting the disadvantage to others.

e. Highway:

Highway is the class that holds *Vehicles* and manages their mobility. We will discuss *Highway's* physical properties, *Vehicle* management tasks, and how users can control vehicles on the highway in order to customize simulations. There are several *Vehicle* management functions that *Highway* performs. *Highway* can automatically create *Vehicle* objects with certain parameters, automatically insert these created objects into lanes, and move each *Vehicle* according to its mobility and lane change models. *Highway* stores each lane as a list structure. When a *Vehicle* object is added to *Highway*, it is inserted in its proper place according to its lane, direction, and *x* position. *Highway* and a reference to a *vehicleID* (set to 1 initially). Any manually-created *Vehicles* should use and increment this *vehicleID* so that all objects will have unique IDs. Note that any manually-created *Vehicles* will be controlled by *Highway* according to the *Vehicle's* mobility model.

B. VanetMobiSim:

VanetMobiSim is an extension to CanuMobiSim [6], a generic *user mobility* simulator. CanuMobiSim is a platform and simulator-independent software, coded in Java and producing mobility traces for different network simulators, including *ns-2* [7], *QualNet* [8], and *GloMoSim* [9]. It provides easily extensible mobility architecture, but, due to its general purpose nature, suffers from a reduced level of detail in specific scenarios. VanetMobiSim is therefore aimed at extending the vehicular mobility support of CanuMobiSim to a higher degree of realism. In this section, we outline the structure and characteristics of VanetMobiSim and detail the resulting vehicular mobility support.

a) Macro-mobility Features:

When considering macro-mobility we not only take into account the road topology, but also the road structure (unidirectional or bidirectional, single- or multi-lane), the road characteristics (speed limits, vehicle-class based restrictions) and the presence of traffic signs (stop signs, traffic lights, etc.). Moreover, the concept of macro-mobility also includes the effects of the presence of points of interests, which influence movement patterns of vehicles on the road topology.

b) Micro-Mobility Features:

The concept of vehicular micro-mobility includes all aspects related to an individual car's speed and acceleration modeling. The micro-mobility description plays the main role in the generation of realistic vehicular movements, as it is responsible for effects such as smooth speed variation, cars queues, traffic jams and overtaking. Three broad classes of micro-mobility models, featuring an increasing degree of detail, can be identified depending on whether the individual speed of vehicles is computed i) in a deterministic way, ii) as a function of nearby vehicles behavior in a single lane scenario, or iii) as a function of nearby vehicles behavior in a multi-flow interaction (i.e., urban) scenario. CanuMobiSim provides implementations for models belonging to the first two classes. The *Graph-Based Mobility Model* (GBMM) [10], the *Constant Speed Motion* (CSM) [6] and the *Smooth Motion Model* (SMM) [11] fall into the first category, as the speed of each vehicle is determined on the basis of the local state of each car and any external effect is ignored. They all constrain a random movement of nodes on a graph, possibly including pauses at intersections (CSM) or smooth speed changes when reaching or leaving a destination (SSM). The movement is random in a sense that vehicles select one destination and move towards it along a shortest-length path, ignoring (and thus possibly overlapping) other vehicles during the motion. While these models may work for isolated cars, they fail to reproduce realistic movements of groups of vehicles.

C. Validation:

In this section, we validate from the study of IDM/MOBIL in ns-3 [3][4]. The first step is to validate that the functions *Model*::*CalculateAcceleration()* and *LaneChange:: CheckLaneChange()* produce output correctly in comparison with Treiber's formula, model, and code individually with various input and mobility model settings. The second step is to produce simple traffic in a one lane roadway and compare the vehicle's acceleration, deceleration, velocity, and position at each simulation interval. Finally, we need to show that despite the difference in our design and the logic of *step* function. The first two steps have been performed during code implementation and testing. Here we show the results of the third step of validation. We use Treiber's Java applet to produce traffic on a straight two lane roadway for several traffic inflow rates. We record traffic statistics (simulation time, vehicle type, acceleration, velocity, position, and lane) at two points. Point *A* is the roadway entrance, and point *B* is 500 m from the entrance. We apply the generated traffic recorded at point *A* in Treiber's applet to our ns-3 simulation and record the traffic statistics at point *B*. This is to mitigate the different injection models used by Treiber's applet and our code. We compare the traffic at point *B* in Treiber's applet with the traffic at point *B* in our ns-3 code during a 5 minute simulation. Figure 2 shows the average traffic density over the 500 m as the traffic inflow rate increases and with different desired speeds. The results between the two applications are almost identical. Figure 3 shows the average differences in position and speed between the two applications for each vehicle as it passes point *B*. Again, there is very little difference between the two. The position differences are less than 7 mm, and the speed differences are less than 1 cm/s.

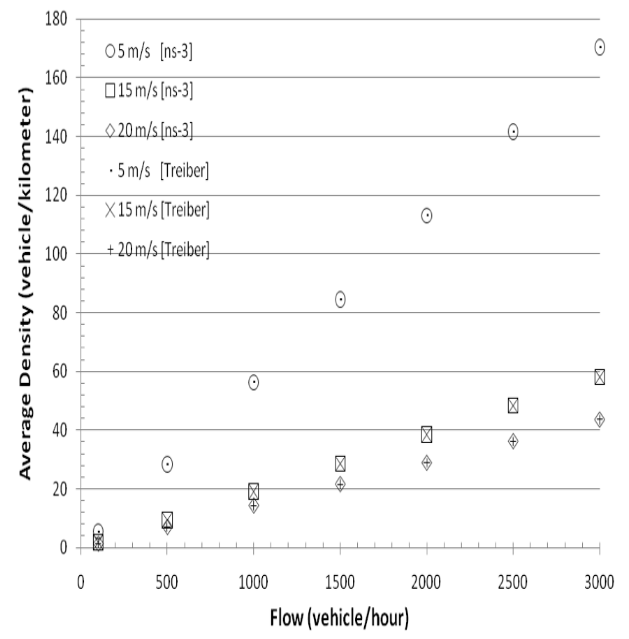

Figure 2: Comparison between average density results of our code in ns-3 and Java applet for different traffic inflow and different desired velocity.

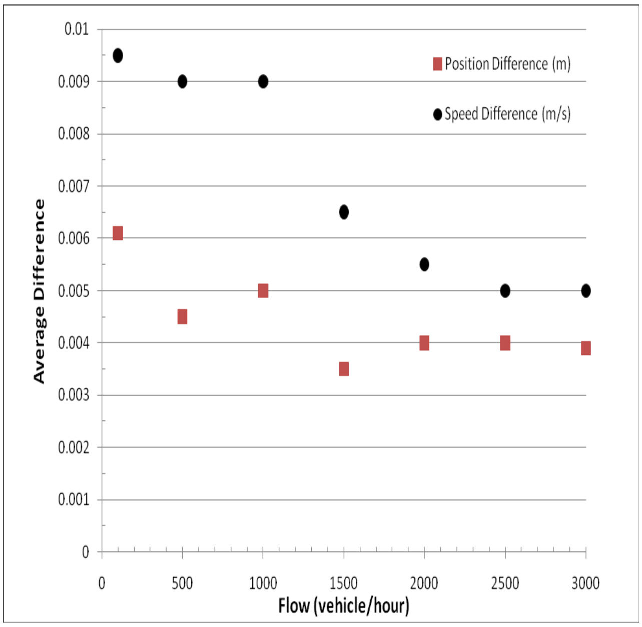

Figure 3: Average difference in position (m) and average speed (m/s) between ns-3 version and Treiber Java applet for different traffic densities.

IV. CONCLUSION AND FUTURE WORK

In this paper, we studied the first implementation of a vehicular mobility model integrated with the networking functions in ns-3. Integrated VANET simulators that include both mobility and network models are essential, allowing network communications our implementation allows for this feedback by triggering an event each time a network message is received and each time vehicle mobility is updated. We presented VanetMobiSim, an extension to the CanuMobiSim user mobility framework capable of producing realistic vehicular mobility traces for several network simulators. We reviewed the macroscopic and microscopic mobility descriptions of CanuMobiSim, Realistic vehicle mobility is achieved through the validated implementation of the IDM car-following model and the MOBIL lane-change model. We introduced the *Highway* class, which not only simulates a straight roadway, but also manages the mobility of all vehicles on the highway. Our implementation also allows the user to take advantage of automatically created and inserted vehicles or to manually insert vehicles at any point along the highway.

In future, we plan to extend our implementation for urban areas (intersections) and add the ability to read in and use detailed maps instead of a single straight highway. We investigate the actual impact of these different traffic phenomena on a vehicular network, so to understand which factors must be considered and which can be neglected for a confident VANETs simulation study. Also, a very important factor when simulating highly mobile networks is the radio propagation model, so the radio signal propagation can hardly be realistic.

V. REFERENCE

[1]. Fiore, M., J. Härri, F. Filali, and C. Bonnet. 2006. VanetMobiSim: Generating realistic mobility patterns for VANETs. Proceedings of ACM VANET. Los Angeles, CA. 96-97.

[2]. Treiber, M., A. Hennecke, and D. Helbing. 2000. Congested Traffic States in Empirical Observations and Microscopic Simulations. Physical Review E, 62(2): 1805–1824.

[3]. Treiber, M. 2006a. Intelligent Driver Model (IDM). Available via <trafficsimulation.de/IDM.html> [accessed April 12, 2010].

[4]. Treiber, M. 2006b. Minimize Overall Braking decelerations Induced by Lane changes (MOBIL). Available via <traffic-simulation.de/MOBIL.html> [accessed April 12, 2010].

[5]. Vanet Mobi Sim, http://vanet.eurecom.fr.

[6]. CANU, http://canu.informatik.uni-stuttgart.de.

[7]. ns-2 http://www.isi.edu/nsnam/ns.

[8]. QualNet, http://www.scalable-networks.com

[9]. GloMoSim, http://pcl.cs.ucla.edu/projects/glomosim.

[10]. J. Tian, J. Haehner, C. Becker, I. Stepanov, K. Rothermel "Graph- Based Mobility Model for Mobile Ad-Hoc Network Simulation",Annual Simulation Symposium 2002.

[11]. C. Bettstetter, "Smooth is Better than Sharp: A Random Mobility Model for Simulation of Wireless Networks", MSWiM 2001, Rome, Italy.

CONFERENCE PAPER
"A National Level Conference on Recent Trends in Information Technology and Technical Symposium" On 09th March 2013
Organized by
Dept. of IT, Jawaharlal Darda Inst. Of Eng. & Tech., Yavatmal (MS), India