# Vehicle Traffic Microsimulator for Ad Hoc Networks Research

Maen M. Artimy, William Robertson, and William J. Phillips

Abstract—Host mobility patterns have an important effect on performance of ad hoc networks. For this reason, many evaluation studies involve analytical or simulation models to synthesize the movement of mobile hosts. The growing interest in investigating the use of ad hoc networks in inter-vehicle communication demands a vehicle mobility model that is capable of creating realistic movements. In fact, there exists a cellular automata (CA) model that is known for its ability to produce such movements.

This paper describes a traffic microsimulator, RoadSim, which implements the basic CA model and extends it in order to generate traffic patterns suitable for use in ad hoc network research. RoadSim is capable of producing traffic that exhibits free flow characteristics, as well as start-stop waves caused by traffic jams. It can also simulate traffic in closed-loop roads or at intersections controlled by traffic signs. We intend to use this tool to evaluate ad hoc protocols in road conditions that, otherwise, cannot be reproduced without more complex and expensive traffic simulators.

Index Terms— ad hoc networks, MANET, vehicle traffic models, inter-vehicle communication, traffic microsimulator.

## I. INTRODUCTION

Mobile, wireless, ad hoc networks (MANET) represent a technology that promises delivery of network access to areas that lack the necessary infrastructure required by other networking technologies. The construction of ad hoc networks among vehicles has been discussed, albeit modestly, since the early stages of MANET development. In recent years, the use of ad hoc networks as an inter-vehicle communication (IVC) technology has gained great interest because of its many potential applications. These include providing support to intelligent transportation systems (ITS) and facilitating Internet access in highways [1], [2].

Given the dominating effect of mobility on MANET performance, most evaluation studies involve a discussion of

some node mobility model. These analytical or simulation models are necessary to generate synthetic movement patterns since real-life traces are difficult to obtain. The literature shows that the results of MANET studies depend heavily on making the appropriate choice of a mobility model [3], [4]. Most studies of MANET protocols focus on people mobility in various environments where mobile nodes change their speed and direction more or less randomly. Vehicle movements are often represented by extending these models to include nodes that move at higher speed (e.g. [4]). In reality, vehicles in road traffic typically follow the road, which restricts their mobility to one dimension. Furthermore, vehicles interact with each other on the road by accelerating, decelerating, and changing lanes. Therefore, representing vehicles as particles moving randomly does not result in realistic traffic patterns.

For this reason, several studies of ad hoc networks use traffic simulators to generate vehicle movement. The studies in [5]-[7], for example, show that realistic representation of vehicle traffic is necessary to evaluate inter-vehicle ad hoc networks in highway environments.

The objective of the work summarized in this paper is to introduce a simulation platform, *RoadSim*, which we designed specifically for MANET research. The platform is based on a traffic model developed by Nagel and Schreckenberg and investigated widely in the literature [8]. This basic model is extended to support the requirements of network simulations. RoadSim avoids the complexities of traffic simulators – which may be designed to reproduce traffic of an entire transportation network - in order to generate accurate traffic patterns in various scenarios such as free flow traffic, traffic in jams, and across intersections.

This paper is organized into six sections. Section II introduces vehicle traffic models and their applications in IVC. Section III reviews the classic traffic flow theory. Section IV contains a high-level description of the implemented traffic microsimulator. Section V summarizes the current studies conducted using RoadSim. The conclusions are provided in Section VI.

# II. VEHICLE TRAFFIC MODELS IN IVC

The representation of vehicle traffic flow has received a great deal of interest because of its applications in urban and non-urban road planning. Experts in transportation networks

M. M. Artimy is a Ph.D. candidate at Dalhousie University, Halifax, Nova Scotia, Canada (e-mail: artimym@dal.ca). W. Robertson is the director of the Master's of Engineering in Internetworking Program at Dalhousie University (e-mail: bill.robertson@dal.ca). W. J. Phillips is the head of the Engineering Mathematics department at Dalhousie University (e-mail: william.phillips@dal.ca).

classify traffic flow simulation models into either macroscopic or microscopic according to the sought level of detail. Macroscopic techniques view vehicular traffic as a fluid. The models are used to describe the aggregate behavior of vehicle traffic by collecting statistics such as density, velocity, and flow. Microscopic models reproduce traffic conditions as they emerge from the interaction of individual vehicles with each other. A brief review of various vehicle traffic models can be found in [9] and [10].

The growing interest in using wireless mobile communication technologies on the road adds a new application to traffic simulation models. The research body may use these models to generate traffic patterns for IVC simulation studies. In such simulations, the use of microscopic traffic models may be of particular importance because of the latter's ability to generate movements of individual vehicles (mobile hosts). For instance, the ability to track the locations of individual hosts in MANETs is necessary to determine the network topology and the connectivity among the hosts.

Microsimulators, simulators based on microscopic traffic models, take a vehicle from its transport network entry point to its final exit. Interactions such as stopping at intersections or changing lanes are carried out in detail to reproduce traffic conditions as realistically as possible. Some microscopic models use various behavioral characteristics of drivers to derive simple rules of interactions among vehicles. In carfollowing models, for example, the rules determine when to accelerate or decelerate in reaction to the vehicle ahead. The models can also decide whether the vehicle maintains a minimum speed by changing lanes whenever necessary.

Car following models have some limitations (e.g. solving a large set of differential equations). To avoid such limitations, focus has shifted to Cellular Automata (CA) systems as alternative models capable of representing individual vehicle interactions as well as producing macroscopic traffic flow metrics. See [11] for a general review of cellular automata.

# III. TRAFFIC FLOW THEORY

The traditional traffic flow theory [12] defines a relationship among three main quantities, vehicle density, flow, and velocity. Vehicles density  $\rho$ , is defined as the number of vehicles counted between two fixed positions on a highway. If vehicles length is ignored, the density is calculated as  $\rho=1/d$ , where d is the average distance between the vehicles.

Car-following models relate vehicles' velocity u to their density by ([12] pp.293),

$$u = \begin{cases} u_{\text{max}} & \rho < \rho_c \\ \lambda \left( \frac{1}{\rho} - \frac{1}{\rho_{\text{max}}} \right) & \rho \ge \rho_c \end{cases}$$
 (1)

This equation indicates that at densities below some critical

density,  $\rho_c$ , a car could travel at maximum speed without interference from other vehicles. At higher vehicle density, velocity declines until vehicles come to a complete stop in jams,  $\rho_{max}$ . The constant  $\lambda(>0)$  measures the sensitivity of vehicle interaction.

The vehicle flow q represents number of vehicles that pass an observer in a certain time. The flow is related to the other quantities by  $q=p \cdot u$ . In fact, since (1) shows that velocity is a function of density, the flow depends only on density,  $q = p \cdot u(p)$ .

## IV. ROAD SIMULATOR (ROADSIM)

The pioneering efforts of Kai Nagel and Michael Schreckenberg in applying CA to vehicle traffic microsimulation have received a great deal of interest in the literature. This interest is due to the (NaSch) model's ability to reproduce basic phenomena encountered in real traffic [13], [14]. The model was used to develop the Transportation Analysis Simulation System (TRANSIMS) [15].

Our main contribution in this paper is using the NaSch model to design a traffic microsimulator for the purpose of generating synthetic traces of vehicle movements in various traffic scenarios. The implementation is focused primarily on generating vehicles movement in an interconnected network of road segments and translating that movement into a format compatible with MANET simulators.

# A. Traffic Model Description

The basic NaSch model consists of a one-dimensional CA grid of L cells with periodic boundary conditions (ring). The model has  $u_{max}+2$  discrete states; a cell can be either empty, or occupied by exactly one vehicle of velocity  $u=0..u_{max}$ . The velocity represents an integer number of cells the vehicle can travel per time step, provided that there are no obstacles ahead. At the beginning of each time step a set of rules is used to update the velocity and position of all vehicles simultaneously. To add random driver behavior, a vehicle may decelerate (for no obvious reason) with probability,  $p_{noise}$ . References [15] and [16] include a comprehensive discussion of velocity update rules in addition to other rules for lane changing and crossing intersections.

The NaSch model is known to produce realistic results when certain values are used. The cell length  $l_c$ =7.5m represents the average length a vehicle occupies in a traffic jam. This results in a maximum density of  $\rho_{max}$ =1/7.5 $\approx$ 133 veh/km. The maximum velocity is set to  $u_{max}$ =5 cell/s =135 km/h.

#### B. Traffic Flow Characteristics

The flow-density curve in Fig. 1 shows that the flow peaks at the critical vehicle density,  $q_{max}=q(\rho_c)$ . If density is increased beyond  $\rho_c$ , traffic jams start to occur and both flow and velocity decline. This realistic phase transition from

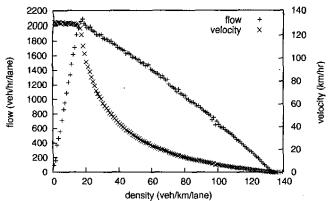


Fig. 1 Flow-density curve (The Fundamental Diagram of Road Traffic) and velocity-density curve. The plots are generated using the output of multiple simulation runs using a closed-loop road segment of length L=1000 cells and various vehicle densities. Each data point represents an average of measurements taken in a 5-cell section over a 3-minute period.

laminar flow to start-stop traffic waves has been well studied in the literature [8], [17].

The velocity curve in Fig. 1 follows the general form of (1) and shows that vehicles travel at free-flow velocity  $u_f$  when density is below  $\rho_c$ . The randomness introduced in the NaSch model by  $p_{noise}$  causes the free-flow velocity to be slightly lower than  $u_{max}$ ,

$$u_f = u_{\text{max}} - p_{noise}$$
 (cell/s).

The figure shows that the flow reaches its peak at  $q(\rho_c)\approx 2100$  veh/hr/lane, where the critical density is  $\rho_c\approx 17$  veh/km/lane. The free flow velocity approaches  $u_f\approx 129$  km/hr.

Other realistic traffic behavior demonstrated by the NaSch model includes constant headway time, instant braking, and delayed acceleration [15]. RoadSim implements lane-changing rules that create equal vehicle density in all lanes. Other sets of rules can create traffic that always prefers, say, the right lane [16] or changes the preference according to a travel plan [15]. Although, the maximum speed is fixed for all vehicles, this can be easily changed by allowing vehicles to have different maximum speeds according to their type (car, truck, etc.) or to the type of road.

#### C. RoadSim Extensions

Some extensions are needed to adapt the original traffic model to the requirements of network simulations. These extensions do not change the principles of the CA model.

#### 1) Road Geometry

Tracking the spatial coordinates of mobile hosts has a significant importance in the operation of a MANET simulator. The simulation uses this information to determine, for example, if a mobile host can establish a link with another host within its transmission range. The coordinates are also important for the operation of some protocols such as geocast routing protocols.

RoadSim maps each cell in the CA grid into a point (x, y) in a 2-D plane. The distances in the x- and y-axis are calculated using mapping functions,

$$x = X(k, n_l)$$
  $k = 0,1, \cdots$   
 $y = Y(k, n_l)$   $n_l = 0,1, \cdots$  (2)

where k and  $n_l$  are the cell and lane index, respectively. The mapping functions determine the geometric shape of the road.

The task of evaluating the mapping functions is handled in a separate module in RoadSim. This approach allows the use of a single run of the CA model to create roads of different geometrical shapes. Moreover, the same approach allows bidirectional highways to be created without modifying the NaSch model. Since vehicles travel in each direction independently, RoadSim can be used to simulate each direction in two runs, perhaps with different set of traffic parameters. The output from each run is mapped into lanes moving in opposite directions.

#### 2) Node IDs

Vehicles in the CA model are anonymous entities since only their travel pattern is significant. Network simulators, on the other hand, use vehicle IDs to track the movement of mobile nodes in edition to network addressing. Therefore, a vehicle ID list is added to the data structure used by RoadSim. The simulator maps the vehicle ID to its position (cell number and lane) in the road during the simulation. The IDs are used when exporting the output to different formats (more details later).

# 3) Time-step Resolution

RoadSim advances time in fixed increments. Therefore, exporting vehicle movement to a network simulator (that advances time based on the next event) may require a change in the time advancement mechanism. A higher time resolution is achieved by producing outputs at smaller time increments than 1s. RoadSim can also generate output only when an event occurs such as a vehicle changing velocity or lane. This approach results in a coarser time resolution.

Time-step resolution should to be increased for other reasons as well. Smooth acceleration/deceleration can be achieved, if required, by increasing/decreasing speed gradually at time steps smaller than 1s. Increasing the time resolution is also required when mapping the CA grid into curved shapes of small radii.

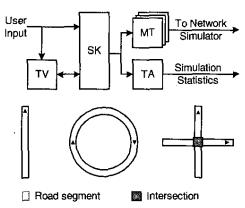


Fig. 2 a) Components of RoadSim, b) building blocks can be used to create traffic in open-loop, closed-loop, and intersection scenarios.

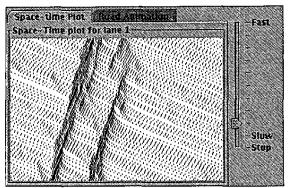


Fig. 3 The space-time plot. The dots represent vehicle locations along a road segment. As time advances (vertically from top to bottom) vehicles move at various velocities. The dark clusters of dots represent traffic jams where vehicles come to complete stop.

### D. Program Components

The simulator is implemented in Java language to take advantage of Java's many capabilities. Fig. 2a illustrates the modular structure of RoadSim. The arrows indicate data flow between various modules of the microsimulator. The following paragraphs describe the function of each module:

A Simulation Kernel (SK) combines instances of two Java classes, a unidirectional road segment and an intersection. These two classes represent the building blocks needed to create various traffic scenarios. Road segments are replicas of the NaSch model but they can have various lengths, number of lanes, and initial number of vehicles. The intersection can be used to connect up to four road segments. Traffic entering the intersection can be either controlled by a Stop or Yield sign, or uncontrolled (no sign). Fig. 2b shows how to use road segments and intersections to create different traffic scenarios.

The SK module initiates the simulation by reading the user's input, and then it advances simulation time by updating the status of all road segments and intersections. The output, in the form of cell states, is supplied to the other modules.

Movement Translators (MT) convert the CA configuration to traces of vehicle movements by evaluating the mapping functions (2). The result is movement script file that can be executed by a MANET simulator.

Currently, there are three MT modules. The first produces traffic movements in a binary form. Among the data included are vehicle position (cell number and lane), ID, and speed. The other two modules map a closed-loop road segment into a straight road and a ring-shaped road, respectively. Both modules produce vehicles movement as an OTcl script that can be used by ns-2 (see next section).

A Traffic Analyzer (TA) taps on a road segment and collects vehicle density, velocity, and flow statistics. The main purpose of this module is to validate the simulation by comparing traffic parameters against the output of other traffic simulators or real-life data. The TA module can be attached to several road segments to collect statistics from different parts of the simulated scenario simultaneously.

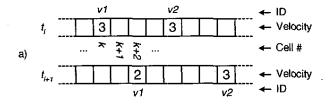
A Traffic Viewer (TV) offers a limited graphical interface to users. The user may choose to invoke the Viewer to display a number of plots and animations. These visuals can be used for demonstration purposes and as an additional validation tool to ensure the correctness of the simulation. Among the produced graphics is the space-time plot shown in Fig. 3. The plot shows the formation of traffic jams when the average density is around  $\rho_c$ . The user can control the update frequency of the simulation when the TV is used.

# E. Compatibility with ns-2

Network Simulator v2, ns-2, is commonly used among the research community for studies of MANETs and other wireless networks [18]. Ns-2 uses two mechanisms to support mobile node movement. Mobile nodes can be set to move randomly or under a program's control. In the latter case, the starting position of mobile nodes and their subsequent destinations and speeds must be set explicitly.

RoadSim generates nodes movement for ns-2 by translating the states of cells at the end of each time step  $t_i$ . As an example of this process, the CA grid in Fig. 4a shows that at the end of time step  $t_i$ , vehicle, vI, has moved to cell k, with speed u=3 cell/s. At  $t_{i+1}$ , vI reduced its speed to u=2 cell/s and moved to cell k+2. The states of vI are translated into two ns-2 OTcl commands shown in Fig. 4b. The first command uses the state of vI at  $t_i$  to set its initial position in the x- and y-axis (the z-axis must be set to zero). The second command moves the vehicle at time  $t_i$  towards a destination point  $(X(k+2, n_L), Y(k+2, n_L))$  at speed of 2x7.5 m/s. The node arrives at destination at  $t_{i+1}$ .

In ns-2, the number of nodes must be constant throughout the simulation, i.e. nodes cannot be created or removed during the simulation (they can be activated/deactivated, however). This works well with closed-loop road simulations where the number of vehicles is constant and they move in a confined space. For open-loop simulations, one possible approach is to initiate the simulation with a large number of vehicles then activate (deactivate) them once they enter (exit) the simulation area.



\$v1 set X\_ < X(k,n<sub>L</sub>)> \$v1 set Y\_ < Y(k,n<sub>L</sub>)> \$v1 set Z\_ 0.0

\$ns at < t > \$v1 setdest  $< X(k+2,n_i) > < Y(k+2,n_i) > < V(2) >$ 

Fig. 4 a) The CA grid at time steps  $t_i$  and  $t_{i+1}$ , b) The corresponding OTcl instructions for ns-2

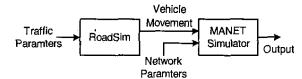


Fig. 5 Using RoadSim in a 2-stage simulation

#### V. APPLICATIONS

We propose to use RoadSim in a 2-stage simulation model shown in Fig. 5. The figure shows that simulating vehicle movement and communication networks are independent. Vehicle traffic input parameters such as length of road segments, vehicle density, and output format are entered in RoadSim. Vehicle movement file is supplied to a network simulator along with input that specifies MANET parameters such as number of connections, traffic type, bandwidth, etc.

Currently, RoadSim is applied in some studies aimed at evaluating various aspects of inter-vehicle ad hoc networks. The objective of the first study is to identify the effects of various factors (e.g. vehicle density, velocity and distance) on the ability to maintain an active link between a pair of communicating vehicles traveling on a highway [19].

RoadSim is also used to study connectivity in inter-vehicle networks. Simulations are used to determine the minimum transmission range in networks formed in 1- and 2-lane roads, bi-directional roads, and intersections.

Preliminary results indicate that communication links break frequently because of vehicle movement. Consequently, there is a large variance in connection periods. The objective of a future study is to identify the ability of some ad hoc routing protocols to react to these frequent breaks in connection. Ns-2 is used to evaluate features such as message buffering and route redundancy.

#### VI. CONCLUSIONS

This paper introduces an implementation of a lightweight traffic microsimulator that is designed to produce realistic vehicle traffic movements for ad hoc networks research. RoadSim is based on the NaSch model that produces realistic traffic behavior. This is of particular importance to MANET research since it allows the evaluation of these networks under various traffic conditions.

In addition to implementing the traffic model as its kernel, RoadSim extends the basic model by assigning IDs to vehicles and tracking them throughout the simulation. RoadSim also maps the CA grid to points in a 2-D plane. This mapping offers flexibility in changing the geometry of road segments to comply with any network simulation requirements. Moreover, RoadSim translates vehicle movements into commands that can be executed by network simulators. Currently, traffic in a closed-loop simulation can be translated to ns-2 API as a straight or ring-shaped road.

Some expected enhancements to RoadSim include adding more MT modules, signalized intersections, variable vehicle velocities, and more advanced graphical user interface. The simulator can include also several types of vehicles such as trucks or buses, and other types of traffic, such as a public transit system, where buses have regular routes and stops.

In addition to simulating vehicle traffic, Cellular Automata models can be used to generate realistic movement patterns for other types of ad hoc networks. These may include pedestrians walking down a busy street, robots exploring a distant planet, or a sensor network collecting surveillance data.

#### REFERENCES

- Mesh Networks inc., "Wirelessly connecting the DOT's; Mesh-enabled solutions for intelligent transportation systems", Available: http://www.meshnetworks.com.
- [2] S. M. Cherry, "Broadband: A Go-Go", *IEEE Spectrum*, vol. 40, no. 6, pp. 20-25, June 2003.
- [3] T. Camp, J. Boleng, V. Davies, "A survey of mobility models for ad hoc network research," Wireless Communications & Mobile Computing, vol. 2, no. 5, pp. 483–502, August 2002.
- [4] C. Bettstetter, "Mobility modeling in wireless networks: categorization, smooth movement, and border effects," ACM SIGMOBILE Mobile Computing and Communications Review, vol. 5, no. 3, pp. 55-67, 2001.
- [5] Z. D. Chen, H. Kung, D. Vlah, "Ad hoc relay wireless networks over moving vehicles on highways," *International Conference on Mobile Computing and Networking*, pp. 247-250, October 2001.
- [6] H. Füßler, M. Mauve, H. Hartenstein, D. Vollmer, M. Käsemann, "A comparison of routing strategies in vehicular ad-hoc networks," *Reihe Informatik*, March 2002.
- [7] M. Rudack, M. Meincke, K. Jobmann, M. Lott, "On traffic dynamical aspects of inter vehicle communications," *IEEE Semiannual Vehicular Technology Conference*, October 2003.
- [8] K. Nagle, M. Schreckenberg, "A cellular automaton model for freeway traffic", Journal de Physique I France, vol. 2, pp. 2221-2229, 1992.
- [9] J. Taplin, "Simulation Models of Traffic Flow," Available: http://www.esc.auckland.ac.nz/Organisations/ ORSNZ/conf34/PDFs/Taplin.pdf
- [10] A. Klar, R. Kühne, R. Wegener, "Mathematical models for vehicular traffic," Available: http://citeseer.nj.nec.com/201102.html.
- [11] P. Sarkar, "A brief history of cellular automata," ACM Computing Surveys, vol. 32, no. 1, pp. 80-107, March 2000.
- [12] R. Haberman, Mathematical Models, Prentice-Hall, Englewood Cliffs, New Jersey, 1977.
- [13] H. Emmerich, E. Rank, "An improved cellular automaton model for traffic flow simulation," Available: http://citeseer.nj.nec.com/15897.html.
- [14] R. Barlovic, L. Santen, A. Schadschneider, M. Schreckenberg, "Metastable states in cellular automata for traffic flow," *The European Physics Journal B*, vol. 5, pp. 793-800, 1998.
- [15] K. Nagel, P. Stretz, M. Pieck, S. Leckey, R. Donnelly, and C. Barrett, "TRANSIMS traffic flow characteristics," Los Alamos National Laboratory report LA-UR-97-3531, March 1999.
- [16] M. Rickert, K. Nagel, M. Schreckenberg, and A. Latour, "Two lane traffic simulations using cellular automata," *Physica A*, vol. 231, no. 4, pp. 534-550. October 1996.
- [17] D. Jost, K. Nagel, "Probabilistic traffic flow breakdown in stochastic car following models," Available: http://citeseer.nj.nec.com/581651.html
- [18] K. Fall, K. Varadhan, The ns Manual, Available: http://www.isi.edu/nsnam/ns/ns-documentation.html
- [19] M. Artimy, W. Robertson, W. Phillips, "Connectivity in inter-vehicle ad hoc networks," (to be published) Canadian Conference of Electrical and Computer Engineering, May 2004.