AN INTEGRATED VEHICULAR AND NETWORK SIMULATOR FOR VEHICULAR AD-HOC NETWORKS

Cristian Gorgorin, Victor Gradinescu,
Raluca Diaconescu and Valentin Cristea
"Politehnica" University Bucharest
Computer Science Department
313 Splaiul Independentei Bucharest Romania
E-mail: {cristig, victor, ralucad}@egov.pub.ro,
valentin@cs.pub.ro

Liviu Iftode
Rutgers University
Computer Science Department
110 Frelinghuysen Road
Piscataway, New Jersey, USA
E-mail: iftode@cs.rutgers.edu

KEYWORDS

Intelligent Transportation Systems, Vehicular Ad-Hoc Networks, simulation, discrete-event, integration, vehicular mobility, wireless communication, network

ABSTRACT

Vehicular ad-hoc networks (VANETs) form when vehicles are equipped with devices capable of short-range wireless communication. Accurate simulation of VANETs is a challenging task, requiring both a vehicle mobility model and a network simulator. Although separate simulators exist, integrating them is difficult. We have developed an integrated simulator, based on studied, validated models. We argue that our simulator can be used for the studying of a large range of VANET protocols and applications, which would be very difficult to study by using other tools.

INTRODUCTION

Vehicle-to-vehicle communication is a very challenging topic in recent years. Vehicles equipped with devices capable of short-range wireless connectivity can form a particular mobile ad-hoc network, called a "Vehicular Ad-hoc NETwork" (VANET). The existence of such networks opens the way for a large range of applications. We consider that two of the most important classes of such applications are those related to route planning and traffic safety.

Route planning aims to provide drivers with real-time traffic information, which would, in the absence of a VANET, require expensive infrastructure. By contrast, the VANET approach is highly scalable and has very low maintenance costs. Moreover, short-range wireless communication technologies (such as 802.11) have no associated cost, other than the communication device.

Safety applications involve disseminating urgent information, which is unavailable in the driver's field of view, or is difficult to notice for reasons such as fog or other vehicles obstructing the line of sight. For instance, a lot of accidents occur in foggy conditions, because drivers notice too late that some kind of incident has occurred in front of them. Safety at intersections could also be enhanced, because the risk of collisions could be detected in advance and the driver could be warned seconds before what would otherwise be an imminent accident.

Most applications to be deployed on top of a VANET require some sort of data-dissemination model. This is a challenging problem, due to the unique characteristics of a VANET. Such a network has a very high degree of nodes' mobility and a very large scale. Network partitioning occurs frequently, making end-to-end communication impossible at times. Several studies (Blum et.al. 2004) show that the performance of classical, topology-based routing protocols in vehicular networks is poor, due to the extremely high mobility of the nodes.

The evaluation of VANET protocols and applications could be made through real outdoor experiments, which should involve a large number of nodes, in order to obtain significant results. However, performing such large-scale experiments is extremely difficult. Therefore, simulation is an indispensable tool.

The simulation of a VANET requires two different components: a network simulator, capable of simulating the behavior of a wireless network, and a vehicular traffic simulator, able to provide an accurate mobility model for the nodes of a VANET. Recent studies (Choffnes and Bustamante 2005) have proven that the vehicular mobility model is very important, and in order to obtain relevant results, it should be well integrated with the wireless network model. The use of an inaccurate mobility model, like the popular random waypoint model (which may work for some mobile ad-hoc networks, but is definitely not an accurate representation of mobility in a VANET), can lead to erroneous results (Choffnes and Bustamante 2005) (Saha and Johnson 2004).

We have developed a simulation tool, comprising the two previously mentioned components: a microscopic traffic simulator, and a wireless communication model. We have also implemented a graphical user interface, using OpenGL for Java (JOGL), which proved useful in some phases of the simulation experiments, but which can be disabled in order to shorten the simulation time.

The remainder of the paper is organized as follows. In the following section we present related work in the area of vehicular networks simulators, along with the motivation for developing our integrated simulator. Another section presents the simulator we have developed. We then show evaluation results and we briefly present applications which could be studied using our simulator. We conclude in the final section.

RELATED WORK

Simulating a vehicular network involves two different aspects. First, there are issues related to the network, such as medium access control, signal strength, propagation delays. Network simulators, like "The Network Simulator – ns-2" (http://www.isi.edu/nsnam/ns) and Jist/SWANS (http://jist.ece.cornell.edu/index.html), cope with such issues. However, a general-purpose wireless network simulator is by no means enough for an accurate simulation of a vehicular network. Nodes in a wireless network usually move according to the random-waypoint model. This means they have an origin and a destination and move towards the destination. But vehicles only move along roads and that is a very particular situation. Furthermore, real vehicles move according to very particular traffic models, due to the street topology, intersections, traffic regulations and drivers' behavior. That takes us to the second very important aspect of a vehicular network simulator, which is using a mobility model as close as possible to real vehicular mobility.

Vehicular traffic simulators can be classified in macroscopic and microscopic simulators. Macroscopic simulators deal with global measures, like traffic flow, while microscopic simulators take into account the movement of each particular vehicle.

There are a lot of commercial vehicular traffic simulators. They have not been designed especially for vehicular computing. They are primarily used to study traffic, in order to validate projects, like building a new road, or a new tram line, or for designing effective traffic signals.

An example of a commercial vehicular traffic simulator is VISSIM (http: // www.english.ptv.de / cgi-bin / traffic/traf_vissim.pl). It is a microscopic simulator and implements driver behavior models, like car-following or lane changing. According to its producers, it is used in over 70 countries. An integrated simulator was developed by a team at Northwestern University. It is based on an original vehicular traffic model, called Street Random Waypoint (STRAW). Their simulator is implemented on top of JiST/SWANS, and it is free and open-source (Choffnes and Bustamante 2005). The authors have used the simulator in order to prove that studying routing protocols for a vehicular network without an accurate vehicular traffic model is a wrong approach. In this respect, they compared results obtained with the Random Waypoint model (which is a very inaccurate representation of a vehicular network) with results obtained with the STRAW model. Their experiments clearly indicate that using the Random Waypoint model will not produce accurate results for a vehicular network.

However, we believe the mobility model implemented in existing simulators (Choffnes and Bustamante 2005) (Saha and Johnson 2004) (Mangharam et.al. 2005) is not a sufficiently accurate representation of real vehicle mobility. Thus, the simulator of Saha and Johnson uses real maps, in the TIGER format (*http://www.census.gov/geo/www/tiger*) and vehicles move along the streets. Each vehicle moves completely independent of other vehicles, with a constant speed randomly chosen. Multi-lane roads or traffic control systems are not taken into consideration. Other authors (Mangharam et.al. 2005) make the same oversimplifying assumptions and do not consider multi-lane roads or car-following models. The mobility model of Choffnes and Bustamante is more complex. It also uses TIGER files, and considers car-following models. The motion of a vehicle is influenced by the preceding vehicle. The authors also implement traffic control systems: timed traffic lights and stop signs. However, multi-lane roads are not taken into consideration.

Furthermore, the majority of VANET applications imply that vehicles react to messages. For instance, if a driver receives a message saying that the road ahead is congested, that driver will change its route. In order to study such reactions, combining an existing vehicular traffic simulator with an existing wireless network simulator is not possible. An integrated simulator is needed.

Based on these aspects, we have chosen to develop a VANET simulation tool, integrating vehicular mobility and wireless transmission simulator.

DESCRIPTION OF THE SIMULATOR

Architecture

The VANET simulator we have developed is a discrete event simulator. The simulation time advances with a fixed time resolution after executing the application code for the current simulation time. More specifically, at every moment of the simulation time, all the current events are pulled from a queue of events, and handled in a random order.

The events queue can hold three types of events: *send*, *receive* or *GPS*. A *send* event for a specified node triggers the calling of the node's procedure responsible for preparing a message. It also schedules the corresponding *receive* event(s) for the receiver(s) the simulator decides to deliver the message to, according to the network module. The *receive* event is associated either with a node, or with a group of nodes (to which the message is broadcasted). Its action is to call the appropriate handler in each of the receiving nodes. The *GPS* event is scheduled at a regular time interval for each node, in order to simulate the way a real VANET application collects GPS data periodically.

The mobility module updates periodically the position of each node representing a vehicle, according to the vehicular mobility model. This model takes into account vehicle interactions (passing by, car following patterns etc), traffic rules and the behavior of different drivers.

The main advantage of this architecture is that the simulator can execute (or emulate) the code of a real vehicular application without significant changes, by using the interface described above. Figure 1 shows the general view of this simulation environment.

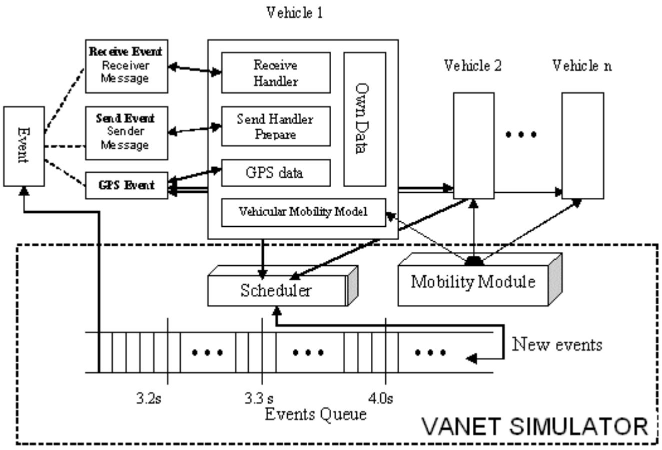

Figure 1 : Simulator architecture

Mobility Model

Maps

A digital map is required in any kind of VANET application. Each vehicle which is part of the system should have such a digital map. For our simulator, we have chosen to use TIGER files, which are freely-available, real digital maps of the USA (http://www.census.gov/geo/www/tiger). The TIGER files contain detailed geographical information about all the roads in a region, from large highways to small streets. The data they contain come in the form of geographical coordinates (latitude, longitude) for the roads. Thus, for every road, the TIGER files specify its end points, along with as many intermediary points as needed, depending on the road's shape. Furthermore, for each road, a "class" information is given (whether it is a small street, a local road, a State Route, an Interstate Highway and so on).

However, the TIGER database unfortunately lacks other traffic-specific information, like the number of lanes, or traffic control systems (traffic lights, yield or stop signs). We believe that a mobility model which does not take multiple lanes or traffic control systems into consideration is not realistic enough; therefore we have added some extra information, based on simple heuristics and based on the road class information included in the TIGER files. Some of the rules we have used include more lanes for higher class roads, yield or stop signs for lower class roads, traffic lights between equally important roads, longer green period for the road with the higher number of lanes and so on. In the future, we can probably expect such detailed traffic information to be contained in real digital maps.

Microscopic Traffic Simulator

A traffic simulator which takes into account the actions of each individual vehicle is a microscopic simulator, as opposed to macroscopic simulators, which describe the evolution of traffic using global measures, like flow or traffic density. Macroscopic simulators can be used to better understand the traffic dynamics and to better design traffic-related facilities (traffic lights, number of lanes, lane closures and so on). However, a much higher level of detail is necessary for the study of a vehicular network; therefore we have developed a microscopic traffic simulator. It is based on the driver behavior model developed by Wiedemann (1974, 1991). The same model is used in the commercial traffic simulator "VISSIM". Like many other vehicular traffic simulators, VISSIM's purpose is modeling and forecasting vehicle traffic flow, for decisions like adding a new lane, studying the impact of lane closures on traffic, building an overpass and so on. Such simulators are difficult to integrate with network simulators, especially since most of them are commercial products.

Next, we briefly describe the driver behavior model we have implemented, which is based on the idea developed by Wiedemann, and further studied by Fellendorf and Vortisch (2000). The basis assumption is that a driver can be in one of four modes: free driving, approaching, following or braking.

Free driving means there is no influence from preceding vehicles on the same lane. In this situation, the driver will seek to obtain and maintain a desired speed. The desired speed and the acceleration depend on the driver personality, and on the road characteristics.

In the "approaching" mode there is a slower, preceding vehicle that influences the driver. In this situation, she/he will apply a deceleration in order to obtain the same speed as the preceding vehicle. The deceleration is a function of the distance between the two vehicles, their speeds, as well as other parameters.

The "following" mode means there is a preceding vehicle, but the speeds of the two vehicles are practically equal. In this situation, the driver will seek to keep the speed constant.

The "braking" mode means there is a slower preceding vehicle, very close in front. In this mode, due to the immediate danger, the driver will apply high deceleration rates.

Figure 2 presents some basic rules to determine the mode that corresponds to a driver. Thus, there are two thresholds, "distance1" and "distance2" according to the notation in the figure. If the preceding vehicle is closer than "distance1" and slower than the current vehicle, then the latter will be in "braking mode". If the slower, preceding vehicle is between "distance1" and "distance2" in front, then the mode will be "approaching", and the current vehicle will gradually decelerate. If the preceding vehicle is further away than "distance2", then it does not influence the current vehicle in any way, and it will be in the "free driving" mode. These thresholds ("distance1" and "distance2") are not constant, but they depend on the driver's personality and on the vehicle's speed.

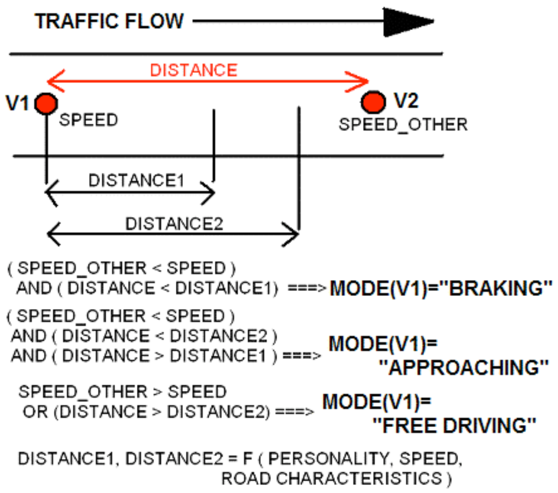

Figure 2 : Driver modes

We have also implemented a lane-changing model, for multi-lane roads. The model we have implemented is based on the lane-usage rules valid throughout most part of Europe. Thus, the usage of the first lane is required, unless it is occupied. It means that a driver will always try to stay on the lower lanes, except when overtaking another slower vehicle. Overtaking on the right side is not allowed. These rules are not valid in city environments, near intersections, where lanes are selected based on the direction the driver intends to follow.

The lane-changing model we have designed and implemented is based on a hierarchy between the four driving modes. Whenever a driver is in a different mode than "free driving", she/he will always check if the higher lane can provide a superior mode. If that is the case, the driver will switch to a higher lane. Similarly, whenever a driver is in a different mode than "braking", she/he will always check if the lower lane provides at least similar conditions. If that is the case, the driver will switch to a lower lane. The order of these checks is important. The higher lane is checked first.

Thus, if a driver uses lane 2 and approaches another slower vehicle, it will first check if lane 3 is empty, and if that is the case it will switch to lane 3 (only if it can safely complete the switch, without interfering with any vehicles approaching from behind). If it had first checked the lower lane, it could have discovered that it is empty and it would have decided to use lane 1 for overtaking the vehicle on lane 2, which is forbidden in most European countries. However, it is not forbidden in the United States, where any lane can be used for overtaking. US traffic could easily be simulated, by making a random decision whether to first check the higher lane or the upper lane when looking for superior driving conditions.

We have also incorporated traffic control systems into our driver behavior model implementation. Thus, the vehicles we simulate are aware of traffic lights, priority roads and "yield" or "stop" signs, and their motion is simulated according to these traffic control systems.

Different driver profiles (aggressive, regular, calm) can easily be modeled by using the numerous model parameters. Each driver class is represented by a certain set of values for the parameters. In order to further differentiate the drivers, there is also a small deviation from the specified values, deviation computed randomly for each driver.

Fellendorf and Vortisch (2000) proved that the model is accurate, by comparing simulated traces with real measurement data taken from a German freeway and from a US freeway. Still, the model is supposed to be accurate not only for freeway conditions, but also for city-like scenarios. To further calibrate and validate our model, we have focused on a simple, yet very frequent, city-like scenario. We considered a typical intersection where vehicles are queued, waiting for a red traffic light to become green (see Figure 3a). We assumed that all vehicles intend to drive forward. Let "FlowPerLane" be the number of vehicles that pass the intersection per second, per lane, during a time period beginning immediately after the light has become green. We consider "FlowPerLane" to be a very important parameter characterizing the motion of vehicles through the intersection, because it is influenced by several parameters of our driver behavior model, like vehicle acceleration, desired distance from the preceding vehicle, or reaction time.

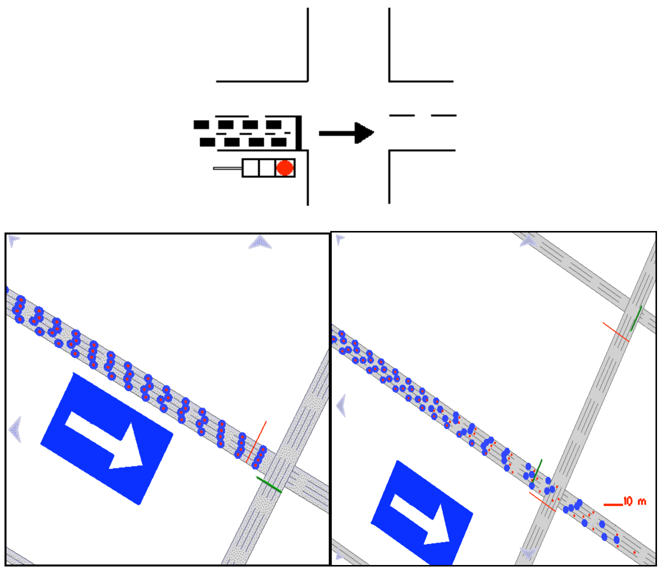

Figure 3 : Typical city scenario and two simulation screenshots

We have chosen the intersections "Piata Victoriei" and "Arcul de Triumf" in downtown Bucharest for measurements. Both intersections meet the above mentioned assumptions. The number of lanes is large (4, respectively 3), and all vehicles are required to drive forward. We measured FlowPerLane by counting passing vehicles during the green phase of the traffic light. We repeated the experiment several times, varying the time frame during which we counted the vehicles, because we suspected there might be a difference between the flow values at the beginning and towards the end of the green phase. The results, however, did not indicate such a difference. We simulated a similar intersection using our driver behavior model. Figure 3 shows screenshots of our simulator's JOGL GUI, taken during the simulation. Figure 3b shows the vehicles still waiting for the red light to become green, while Figure 3c shows the vehicles as they have started passing, as the light has turned green. We have calibrated some of the numerous driver behavior model parameters, based on the real results obtained. Finally, with the calibrated parameters, we have performed several measurements. The measured data and the simulated data are presented in the table in Figure 4.

It is easy to see the similarity between the simulated data and the real situation. The simulated data values have an average of **0.46** and a standard deviation of *0.03*. The measured data values from "Piata Victoriei" have an average of **0.45** and a standard deviation of *0.04*. Finally, the measured data values from "Arcul de Triumf" have an average of **0.47** and a standard deviation of *0.02*.

SIMULATED DATA				REAL MEASURED DATA			
NUMBER OF VEHICLES	NUMBER OF LANES	TIME (SECONDS)	FLOW PER LANE (VEHICLES/SEC)	NUMBER OF VEHICLES	NUMBER OF LANES	TIME (SECONDS)	FLOW PER LANE (VEHICLES/SEC)
47	4	26	**0.45**	PIATA VICTORIEI			
46	4	26	**0.44**	12	1	26	**0.46**
49	4	26	**0.47**	12	1	26	**0.46**
49	4	26	**0.47**	10	1	26	**0.38**
48	4	26	**0.46**	49	4	26	**0.47**
51	4	26	**0.49**	50	4	26	**0.48**
47	4	26	**0.45**	6	1	12	**0.50**
48	4	26	**0.46**	18	4	12	**0.38**
25	4	12	**0.52**	23	4	12	**0.48**
24	4	12	**0.50**	22	4	12	**0.46**
22	4	12	**0.46**	ARCUL DE TRIUMF			
24	4	12	**0.50**	58	3	40	**0.48**
24	4	12	**0.50**	55	3	40	**0.46**
65	4	40	**0.41**	53	3	40	**0.44**
67	4	40	**0.42**	57	3	40	**0.48**
67	4	40	**0.42**	37	3	26	**0.47**
68	4	40	**0.43**	39	3	26	**0.50**
66	4	40	**0.41**	35	3	26	**0.45**
69	4	40	**0.43**	38	3	26	**0.49**

Figure 4 : Comparison of measured and simulated data

Based on the strong similarities between the real and the simulated data, we conclude that the model is an accurate approximation of vehicular mobility, in the above-mentioned, frequently-met, city scenario.

Network Simulator

The network simulator module copes with the delivery of messages from one node to another. It offers a set of network primitives that can be called by the node applications emulated on top of this simulation framework. Of special interest are the MAC and physical layers that determine VANET applications performance (Takai et.al. 2001).

At the physical layer, we use a model with cumulative noise calculation and signal reception based on SNR (Signal-To-Noise) threshold. This means that when a radio receives a signal of a given strength, the noise is calculated as the sum of all the other signals on the channel, and the ratio of the two values is the SNR. The signal can be successfully received if the value of SNR is higher than a given threshold SNRT.

The radio wave propagation can be affected by three independent phenomena: path loss, fading and shadowing (Takai et.al. 2001). The path loss effect is considered to be the most important factor and it reflects the signal power attenuation due to the propagation distance. Our simulator has two signal propagation models: free-space and plane earth two-ray path loss. While the first is an idealized model, the two-ray path loss model considers the effect of earth surface reflection and is more accurate.

Our simulator delivers a message to all the nodes in the wireless range in an optimized way using a local search of nodes. This is possible due to efficient indexing of the map points, using the PeanoKey mechanism (Dashtinezhad et. al. 2004). A PeanoKey is associated with a point in the 2D space, and it is obtained by interleaving the digits of the two coordinates. Thus, the 2D set of points is represented in a one dimensional set. For example the PeanoKey associated with the geographical point at 26.047800 degrees longitude and 44.435348 degrees latitude is 4246403457384080. When the map is being built, a set of sorted PeanoKeys is also computed, corresponding to all the points of the map. Consecutive PeanoKeys in this set correspond to points that are close on the map. In this manner the wireless environment of a node is quickly analyzed, its wireless neighbors are discovered and a map of the radio signal is built.

At the Link Layer, we have implemented the CSMA/CA channel access mechanism which is the base of IEEE 802.11 standard. The basic principles of CSMA/CA are *listen before talk* and *contention*. When a node has to send a packet, it starts by listening the environment and if idle, begins the transmission. If the medium is busy, the node waits for a random amount of time before checking again.

A lot of work has been done to study routing layer protocols in VANETs. Classical topology-based protocols have been proven to perform poorly (Blum et.al. 2004), and location-based approaches have been suggested (Festag et.al. 2004). We have implemented a geographical routing protocol for the routing layer. A node uses geographical information about its neighbors, the origin and the destination of a packet, in order to make forwarding decisions. However, due to the very high mobility and the frequent partitioning of VANETs, no guarantees can be made about reliable end-to-end communication.

Fuel Consumption and Pollutant Emissions Estimation

Estimating fuel consumption and pollutant emissions is an increasingly important matter when studying vehicular traffic. Studying the improvements which VANET applications can bring on these parameters is a possible usage of our simulator. Therefore, we have considered useful to integrate the computing of fuel consumption and pollutant emissions. The model we have implemented is influenced by the work of Akcelik and Besley (2003). Of special relevance to our work, we consider the estimation of the relation between fuel consumption and emissions and the speed and acceleration of the vehicle. The model is simplified to take into account only light vehicles. Thus, based on the motion of vehicles, our simulator's engine accurately computes the fuel consumption and pollutant emissions of each vehicle. Statistics and global measures can easily be obtained.

Traffic Scenarios Generation

We have developed a GUI which can be used to generate traffic scenarios. The user can see a graphical representation of a given TIGER map and can specify flows of vehicles. A flow consists of an entry point, an exit point, a route, and the actual vehicular flow value (in vehicles/hour/lane). The user can also specify how the flow value varies over time.

SIMULATOR EVALUATION

The integrated simulator we have developed is able to simulate around 10.000 network events per second, on a 1.6GHz uni-processor. Although this value is clearly lower than the throughput of the widely-used network simulator ns2 (which can simulate over 60.000 events per second, on a 2GHz uni-processor), it must be noted that our network simulator is integrated with a complex node mobility simulator, responsible for accurately computing the motion of all nodes, every time cycle.

As a basic vehicular computing application to experiment with our simulator, we have chosen TrafficView (Dashtinezhad et. al. 2004). This application assumes each vehicle is equipped with a GPS receiver and a wireless communication device, and has a unique identifier. Periodically, each vehicle broadcasts information about its location. This information can be forwarded further by neighboring vehicles, thus creating a platform where each vehicle is aware of its neighbors.

Our simulator is able to simulate real-time (1 second of simulation in 1 second of real time) the motion of 1000 vehicles, in a complex city-scenario: a square region of 1km by 1km, representing a part of downtown Manhattan, with a large number of intersections and traffic lights (Figure 5). While moving, all the nodes run the simple neighbor-discovery and update protocol, with a 1 second period for the beacons.

We have also successfully performed the simulation of more complex scenarios, involving up to 10.000 vehicles, and we were able to obtain significant results, in spite of the increased simulation time.

The graph in Figure 6 shows how the increase of the number of nodes influences our simulator's performance. The results are based on simulations performed in a highway traffic scenario, with all the nodes running a neighbor-discovery and update protocol, with a 1 second period for the beacon messages. The simulations were performed on a 1.6GHz uni-processor, with 512Mb of memory.

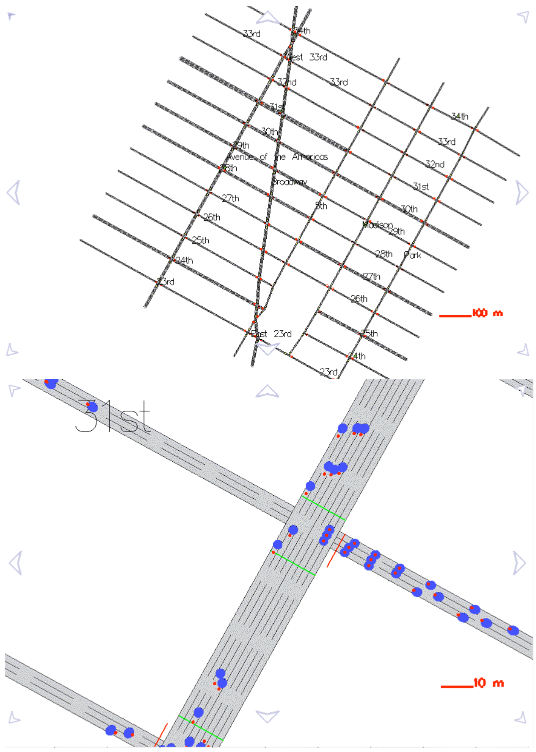

Figure 5 : Simulation screenshots

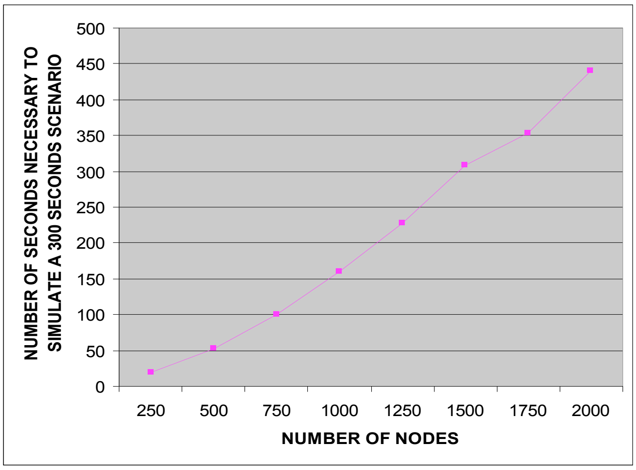

Figure 6 : Simulator performance

As previously described, three main parts can be distinguished in the simulation process: mobility, simulator engine and emulation of the nodes' application. Optionally, a graphical user interface may show simulation details. Obviously, additional time is consumed with display functions and the synchronization mechanisms. The most time consuming part of the simulation is the emulation of the application code, which should be run individually be each node (Figure 7). The TrafficView application has to parse all the incoming messages, update the local vehicle records and create new messages for broadcast. Figure 7 shows that for high densities, when the network is widely connected, messages are propagated easily from car to car and more than half of the simulation time goes on processing the messages received by each of them.

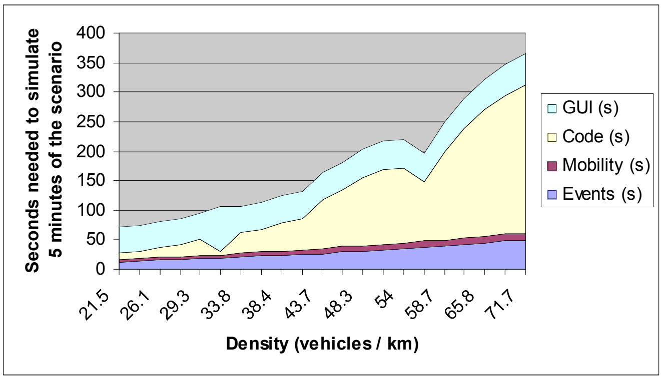

Figure 7 : Time measurements for the simulation process and its components, depending vehicles density. Test scenario: 10 km of highway with a traffic flow varying between 500 and 1500 vehicles/hour/lane.

APPLICATIONS AND FUTURE WORK

A large range of vehicular computing applications can be evaluated using our simulator. We are currently in the process of studying three such applications.

First, we are developing an adaptive traffic lights system in which wireless traffic lights can obtain real-time traffic information by communicating with vehicles. We focus the study on algorithms for efficient traffic control. Our integrated simulator allows us to easily emulate fixed nodes (traffic lights) and the application running on top of them. The vehicles move according to the traffic signals, because of the mobility model. We can easily compare traffic fluency when using different solutions for the traffic lights.

Secondly, we are studying a query-reply protocol. A node could make use of the ad-hoc network in order to obtain real-time traffic information about remote regions. This application makes use of the geographical routing protocol. We want to see in what conditions such an application can work, in the highly mobile environment of a VANET.

Finally, we are studying an application for suggesting best routes to drivers, based on real-time traffic information. By using the ad-hoc network and/or infrastructure, drivers can obtain the best route to a destination, taking into account the current traffic shape.

CONCLUSIONS

The studying of VANET protocols requires efficient, accurate simulation tools. Existing simulators which have not been designed especially for VANETs are difficult to use. We have developed an integrated simulator, comprising a complex model for vehicles mobility, a wireless network simulator and an interface for the emulation of vehicular applications. On top of this simulator we have implemented TrafficView (Dashtinezhad et.al. 2004), an application for information exchange between vehicles.

In this context, we have analyzed the performance of the simulator, and found it can be used to simulate networks of several thousands of nodes, in complex city scenarios, as well as highway scenarios. The simulator allows the evaluation of a large range of vehicular computing applications, which cannot be studied by using existing simulators.

ACKNOWLEDGEMENT

This material is based upon work supported in part by NSF under grants ANI-0121416, CNS-0520123 and CNS-0584475.

REFERENCES

Akcelik R., Besley M. 2003. “Operating cost, fuel consumption, and emission models in aaSIDRA and aaMotion”. *25th Conference of Australian Institutes of Transport Research (CAITR 2003)*

Blum J., Eskandarian A. and Hoffman L.J. 2004. “Challenges of Intervehicle AdHoc Networks”. *IEEE Transaction on Intelligent Transportation Systems*, 5(4):347-351.

Choffnes D.R. and Bustamante F.E. 2005. “An Integrated Mobility and Traffic Model for Vehicular Wireless Networks”. *Proc. of the 2nd ACM International Workshop on Vehicular Ad Hoc Networks (VANET), September 2005.*

Dashtinezhad S., Nadeem T., Dorohonceanu B., Borcea C., Kang P. and Iftode L. 2004. “TrafficView: A Driver Assistant Device for Traffic Monitoring based on Car-to-Car Communication”. *Proceedings of IEEE Semiannual Vehicular Technology Conference*

Fellendorf M. and Vortisch P. 2000. “Validation of the Microscopic Traffic Flow Model VISSIM in Different Real-World Situations”

Festag A., Fussler H., Hartenstein H., Sarma A. and Schmitz R. 2004. “FleetNet: Bringing Car-to-Car Communication into the Real World”. *Proc. Of the 11th World Congress on ITS, Nagoya, Japan.*

Mangharam R. Weller D.S., Stancil D.D., Rajkumar R., Parikh J.S. 2005. “GrooveSim: A Topography-Accurate Simulator for Geographical Routing in Vehicular Networks”. *Proc. of the 2nd ACM International Workshop on Vehicular Ad Hoc Networks (VANET), September 2005.*

Saha A. and Johnson D. 2004. “Modelling Mobility for Vehicular Ad-hoc Networks”. *ACM International Workshop on Vehicular Ad Hoc Networks (VANET)*, 2004.

Takai M., Martin J., Bagrodia R. 2001. “Effects of Wireless Physical Layer Modeling in Mobile Ad-Hoc Networks”. *ACM International Symposium on Mobile Ad Hoc Networking and Computing* (MobiHoc 2001)

US Census Bureau, Topologically Integrated Geographic Encoding and Referencing system (Tiger Line) - *http://www.census.gov/geo/www/tiger.*

Wiedemann R. 1974. “Simulation des Straßenverkehrsflusses”. Schriftenreihe des Instituts für Verkehrswesen der Universität Karlsruhe, Heft 8

Wiedemann R. 1991. “Modelling of RTI-Elements on multi-lane roads”. *Advanced Telematics in Road*

Transport edited by the Comission of the European Community, DG XIII, Brussels (1991).

AUTHOR BIOGRAPHY

CRISTIAN GORGORIN, VICTOR GRADINESCU and **RALUCA DIACONESCU** obtained their Bachelor of Science in Computer Science degrees from "Politehnica" University Bucharest, in the summer of 2006. In the summer of 2005, they were invited by Rutgers University to visit the Computer Science Department, and they worked with Professor Liviu Iftode in the field of vehicular computing. This paper is partly based on work performed during that time. The paper is also based on work performed for their graduation theses, coordinated by Professor Valentin Cristea. Presently, they are all pursuing Master of Science degrees, in the Computer Science department of University "Politehnica" Bucharest.