AUTOMOTIVE NETWORKING SERIES

VGSim: An Integrated Networking and Microscopic Vehicular Mobility Simulation Platform

Bojin Liu, Behrooz Khorashadi, Haining Du, Dipak Ghosal, Chen-Nee Chuah, and Michael Zhang, University of California, Davis

*This research is funded in part by the National Science Foundation under the grant number CMMI *0700383. The authors are solely responsible for the contents of this article.*

ABSTRACT

Simulation is the predominant tool used in research related to vehicular ad hoc networks. In this article we first present the key requirements for accurate simulations that arise from the various applications supported by VANETs, and review the current state-of the-art VANET simulation tools. We then present VGSim, an integrated networking and microscopic vehicular mobility simulation platform. VGSim provides full-fledged wireless network simulation with an accurate traffic mobility model. These two components are tightly integrated and can interact dynamically. We discuss the flexibility of VGSim in adopting different mobility models and also present simulation results that empirically validate the modified mobility model we implemented. We discuss how VANET applications can be easily modeled in VGSim, and demonstrate this using two important applications, Accident Alert and Variable Speed Limit.

INTRODUCTION

Vehicular ad hoc networks (VANETs) are mobile wireless networks formed by vehicles with wireless communication and positioning capabilities. During the last few years, VANET has become a very popular field of research both in academia and in industry. This is both due to the widespread emergence of robust wireless networking and positioning technologies, and, more important, the demand for the next-generation intelligent transportation system to provide both real-time traffic management and commercial services to vehicles on the road.

Due to the nature of mobile wireless communications and the complex dynamics in real vehicle traffic flow, simulation is the primary tool of choice to analyze various applications of VANETs. Sophisticated simulation packages are available for both wireless networks and vehicular traffic flow; however, few of them can fully address the challenging problems that arise from the interdisciplinary nature of VANETs. Therefore, integrating network simulation tools with realistic vehicular traffic simulation packages is necessary. There are different approaches to integrating these two simulation packages. However, if the underlying integrated tool cannot fulfill all the requirements imposed by VANET applications, the results are prone to be erroneous or unrealistic. Therefore, a classification of VANET applications based on simulation requirements is necessary for accurate simulation design.

In general, VANET applications can be classified into the following two categories:

- **Vehicular driver safety and traffic control applications**: These applications need to address the issue of how drivers respond to the control signals disseminated using wireless communication and the resulting change in the topology of the underlying VANET. Typical applications are accident alert, real-time traffic condition update, and any applications that require driver coordination through the VANET.
- **Infotainment Applications**: These applications use VANET as a single- or multihop communication platform, and do not result in dramatic change in the topology of the underlying VANET. Typical applications include Internet access to vehicles, commercial advertisements, and various peer-to-peer applications.

For both of the above classes of applications, a network communication simulation package with full protocol stack support is desired. For vehicular traffic simulation, realistic traffic mobility models are also required. For infotainment applications, simply integrating these two simulation packages by using vehicular traffic traces to determine node movements in network simulation is sufficient. However, for vehicular driver safety and traffic control applications, real-time interactions between the network simulation module and vehicular traffic simulation module are required.

The remainder of the article is organized as follows. In the next section we present different aspects of the most commonly used simulation methodologies in VANET research (trace driven, open-loop, and closed-loop integration). We then present VGSim, an integrated VANET simulation platform that has full-fledged network protocol support, a realistic microscopic vehicular traffic model, and the ability to support real-time interactions between the two modules. In the following section we first present the results that validate the mobility model in VGSim and then discuss VGSim's ability to adopt other mobility models. We then discuss how VANET applications are developed for simulation analysis in VGSim. We then showcase vehicular driver safety applications analyzed using VGSim. The final section concludes this article.

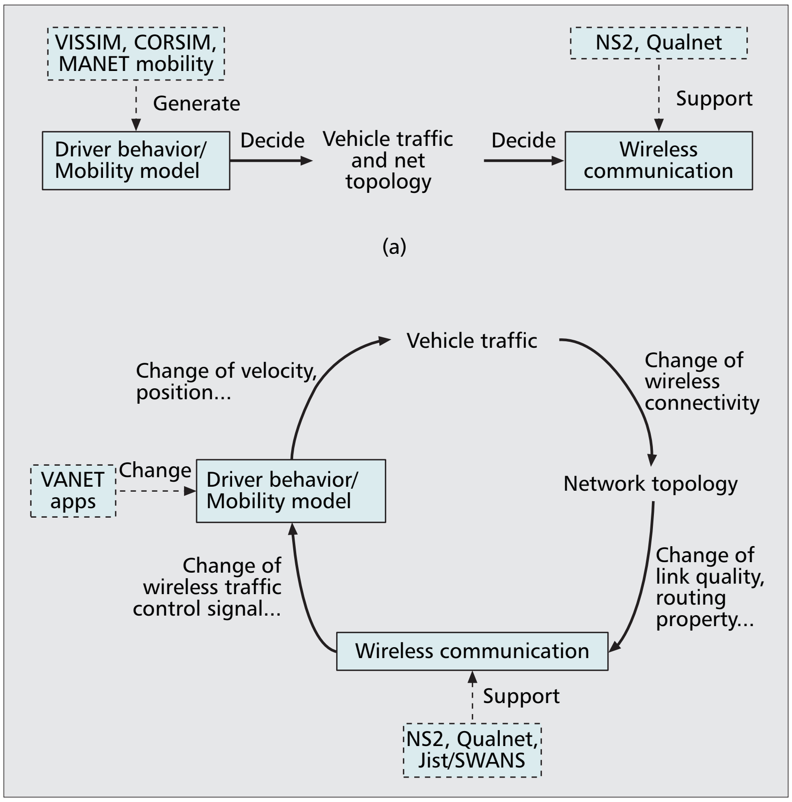
VANET SIMULATION METHODOLOGIES

SIMULATION REQUIREMENTS AND DESIGN

In general, for VANET simulation there are three dimensions in the design space: network simulation, vehicular traffic simulation, and the integration of these two modules. We conducted a survey of VANET research published during the last four years. As our survey shows, besides in-house simulators, running a network simulator with traffic traces generated by a traffic simulator is the main approach for VANET simulation (more details in the next section). The traffic traces specify the vehicle mobility during simulation. We refer to this method as the open-loop integration approach. The key disadvantage of this approach is that it cannot capture the dynamic interactions between the information exchange among the vehicles and/or roadside sensors and the traffic flow.

Another commonly used approach is the closed-loop integration approach. In this approach the traffic simulator is responsible for specifying vehicle movements throughout the simulation process, and the network simulator is responsible for wireless communication. However, signals transmitted using wireless channels could be used as another type of traffic control signal, which can result in a change in the vehicles' mobility. This is especially true for advanced distributed multihop vehicular driver safety and traffic control applications. In these applications driver coordination based on wireless traffic control signals can dramatically change drivers' behavior (accelerating, decelerating, or changing lanes), and therefore results in a traffic flow different from what a traditional traffic simulator could generate. Changes in the traffic flow may imply changes in the topology of the wireless ad hoc network formed by the vehicles, which in turn can have significant impact on the performance of the wireless network. Therefore, a unique requirement for this type of VANET simulation is the ability of capturing the "interactions" between wireless communication and the vehicular mobility model. Figure 1b shows the details of information flow and the interactions of the closed-loop approach.

On the other hand, not all VANET applications require this "interaction" capability in their simulation. Infotainment applications, which only use a VANET as a medium to transmit value added services such as real-time advertisement and Internet access service, do not necessarily affect the underlying topology of the VANET. If data dissemination is the only application of a VANET, the current approach of simple integration of a network simulator and a traffic simulator (the open-loop approach) is sufficient. It is, however, necessary that the adopted network simulator support the entire wireless communication network protocol stack to be able to carry out detailed network performance analysis.



■ **Figure 1.** *Information flow of two VANET simulation approaches.*

In addition, since VANET simulation platforms are needed for evaluating potential safety/infotainment applications, ease of new application development should also be considered in the design. Simulation platforms that adopt the approach of integrating existing traffic and network simulators may encounter complexities in building new VANET applications. This is because it requires the expertise in both simulation packages to build an efficient VANET application. Also, flexibility in adopting different mobility models and performance issues to support large-scale VANET simulations involving hundreds or even thousands of communication nodes (vehicles) are also important factors in the design of the simulation tool.

RELATED RESEARCH ON VANET SIMULATION STUDIES

We conducted a survey of VANET research published during the last four years; due to space limitation, we only highlight the most relevant.

As discussed above, simulation analysis of

Simulation approach	Description	Mobility model	Examples
Open-loop, simplistic mobility model	Network simulator with simplistic MANET or macroscopic models	MANET models or macroscopic traffic models	[9, 10]
Open-loop, trace driven mobility model	Microscopic simulation such as VISSIM generated vehicle traces fed into network simulator (e.g., NS2, QualNet)	Microscopic traffic models	[11–17]
Closed-loop, realistic mobility model	Integrating network communication and vehicular traffic simulation, supporting interaction between the two	Microscopic traffic models	[4, 18–20]

Table 1. *A classification of major simulation approaches in recent VANET research.*

VANETs and related applications requires both communication network and vehicular traffic simulations. For the communication network simulation, our survey shows that the majority of research adopted established network simulators (NS2 [1]/QualNet [2]). For VANET simulations that only considered high-level communication parameters like transmission range, some simple network simulators were used. While Java in Simulation Time (JiST)/SWANS [3] is not as popular as NS2/Qualnet, there are a number of studies on VANETs based on this platform [4]. OMNet++ with INET Framework [5] is another platform used for simulation analysis of wireless communication.

For vehicular traffic simulations, three types of vehicle mobility models are typically used:
- Mobility models used in mobile ad hoc networks (MANETs) and variants
- Macroscopic vehicular traffic models
- Microscopic vehicular traffic models

MANET mobility models (e.g., the Random Way Point model) are not accurate for realistic vehicular traffic simulation, and can considerably degrade the accuracy of the simulation results [4]. Macroscopic traffic models only specify high-level traffic metrics such as vehicle density and flow rate. A microscopic vehicular traffic model, on the other hand, specifies the behavior of each individual vehicle. As a result, microscopic models can generally provide more realistic mobility patterns and detailed statistics of vehicular traffic flow.

For simplicity, a large body of work is based on self-developed macroscopic traffic models. When more realistic microscopic traffic models are used, they are based on either high fidelity traffic simulators such as VISSIM [6], CORSIM [7], and SUMO [8], or simulators developed by the researchers.

Table 1 summarizes the main approaches used in VANET simulation. In the open-loop approach with a simplistic mobility model, established network simulators like NS2 were used for network simulation, and simple vehicular mobility models based on a MANET or simple macroscopic vehicular traffic models were used to generate vehicular traffic flows [9, 10]. Open-loop means the vehicle mobility model is specified at the beginning of the simulation, and underlying attributes of vehicular traffic flow such as headways between vehicles and speed are predetermined and do not change as a result of the VANET application. Figure 1a shows the information flow of the open-loop approach. Specifically, the study reported in [9] describes how to modify NS2 to accommodate this type of VANET simulation. The open-loop approach can also be trace-driven: vehicular traffic traces generated from high fidelity commercial/noncommercial microscopic vehicular traffic simulators such as VISSIM [11, 16], CORSIM [12] or SUMO [15], or empirical traffic traces [13, 14] are used to describe the mobility of vehicles. The traffic traces specify each individual vehicle's movement during the entire simulation. The study reported in [17] is another example of this approach; it adopts a microscopic mobility model (IDM/MOBIL model) to determine the movement of the vehicles and uses OMNet++ [5] to simulate the communication network.

Figure 1b shows the information flow for the closed-loop approach, in which vehicular movements are not predetermined at the start of the simulation. Instead, the mobility model updates the vehicle position, velocity, and lane in real time based not only on the vehicular traffic flow but also on the traffic flow control signal received through wireless communication. The altered mobility of vehicles can in turn affect the topology of the VANET and consequently the performance of the data communication over the wireless network. This closed-loop information flow between the mobility model and wireless network simulation modules cannot be provided by the open-loop approach. Consequently, several studies have adopted this closed-loop approach for simulation analysis [4, 18–20]. In [4] closed-loop integration is achieved by integrating the JiST/SWANS network simulator with Street Random Waypoint (STRAW), a modified version of the Random Waypoint mobility model. This work also demostrates the importance of realistic mobility models for the accuracy of VANET simulation results. It showed that an unrealistic mobility model of the vehicle can dramatically affect the simulation results. The study reported in [18] also directly supports closed-loop interaction between the mobility model and the wireless communication module. It adopts SUMO [8] as the traffic simulator and OMNet++ [5] for wireless communication simulation. The interactions between these two modules are achieved by connecting two simulators through a TCP connection that is used to transfer control commands to SUMO and vehicle position information to the OMNet++ module. In addition, [19, 20] are also integrated VANET simulation platforms based on closed-loop integration between the two simulation modules. The difference between [19, 20] and other closed-loop approaches is that both network and vehicular traffic simulation modules are self-developed by the authors. This, however, makes it difficult to compare with other research based on more well-known simulation packages such as NS-2 and VISSIM.

While all approaches can, to some extent, fulfill the requirements of infotainment applications, only closed-loop approaches are suitable for accurately simulating vehicular driver safety and traffic control applications.

VGSim: Design and Implementation

In order to easily design and analyze different VANET applications through realistic simulation, we developed a highly efficient and flexible VANET simulation platform: VGSim. VGSim is efficient in memory usage and suitable for simulating large-scale vehicular wireless networks. It consists of a network simulator with full protocol stack support, a realistic microscopic vehicular mobility model, and the closed-loop approach to integration. VGSim's network simulation module is based on SWANS [3], a Java-based network simulator. The SWANS network simulator uses JiST, which is an event driven simulation tool [3]. The JiST simulation platform is very efficient; it outperforms existing highly optimized simulation tools in both time and memory usage. A detailed comparison of the performance efficiency of JiST/SWANS compared to other major network simulators can be found in [3]. In fact, the efficiency of JiST/SWANS makes it very suitable for VANET simulation, which may involve hundreds or even thousands of simultaneously communicating nodes.

In VGSim vehicular movements and applications are transformed into events that are processed by the JiST event driven platform. The network simulator and the vehicular traffic model run on a feedback loop that enables the closed-loop interaction discussed in previous sections. Information obtained from the SWANS network simulator is fed into the mobility model and then based on the mobility model, updated antenna positions are determined for the SWANS network simulator. Figure 2 shows the architecture of VGSim. Each entity shown in Fig. 2 has a corresponding class defined in Java. Instances of the `RoadEntity` class represent the road sections and hold multiple `Vehicle` instances during simulation. Each `Vehicle` instance mounts a radio antenna and implements the wireless network communication protocols defined in JiST/SWANS. Each individual object can produce and respond to simulation events generated by itself or other objects. In addition, the SWANS network simulator and vehicular mobility simulator both update a graphical interface that allows network and vehicular mobility parameters to be changed dynamically. Visualization is another important feature for both vehicular traffic and wireless communication simulation, and many vehicular traffic and network simulators have their own visualization packages [1, 6]. Enabling visualization for vehicular traffic and wireless communication at the same time in the same panel is an important feature in VANET simulation, since it can help in visually evaluating the correctness and effectiveness of VANET applications. Figure 3 shows a screenshot of VGSim simulating a four-lane freeway scenario with a roadside node and VANET enabled vehicles communicating with each other. It clearly shows VGSim's visualization capability of overlaying communication traffic on top of vehicular traffic.

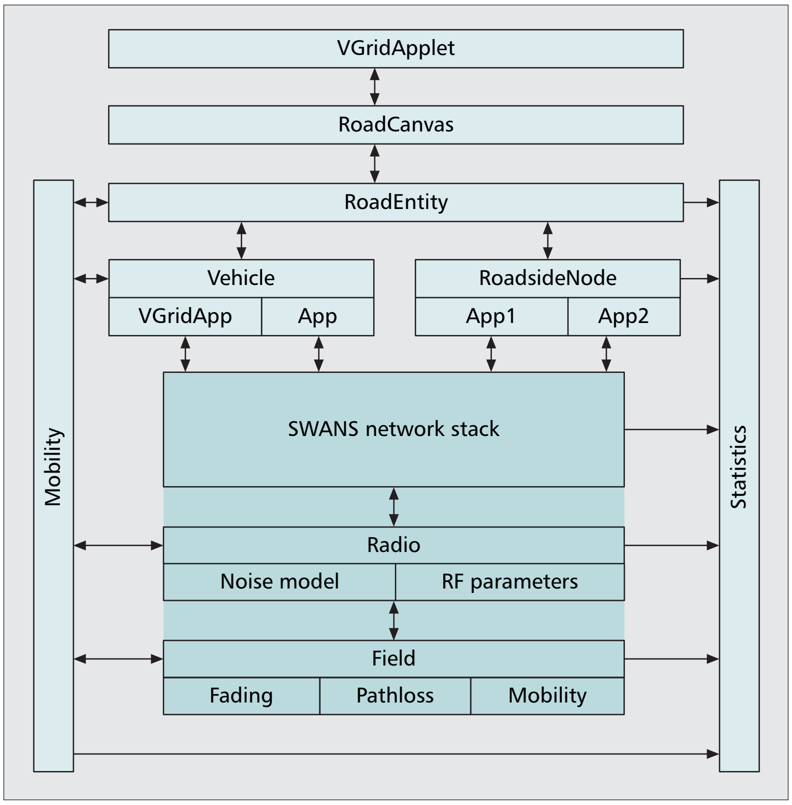


Figure 2. *A block diagram of the VGSim architecture.*

The vehicular mobility module of VGSim is based on the cellular automata (CA) model, which implements a modified version of the Nagel and Schreckenberg (N-S) model [21]. The NS model is a well established CA model in traffic engineering research. However, in the original N-S model, the road is divided into equal-length cells of 7.5 m, and each vehicle occupies one cell. The simulation time granularity is 1 s; hence, new vehicle positions are calculated every second using the N-S model. In order to more accurately reflect real-world traffic, we modified the original N-S model with finer spatial and temporal resolution, based on the study reported in [22]. Furthermore, we also added lane-changing capability into our mobility model. We discuss the validation of our mobility model in the next section.

The SWANS network simulator provides full network protocol support especially for mobile wireless communication. At the application layer, SWANS provides the standard application network interfaces. It includes both UDP and TCP protocols at the transport layer. We also implemented a simple position-based routing protocol, which leverages the GPS devices in the

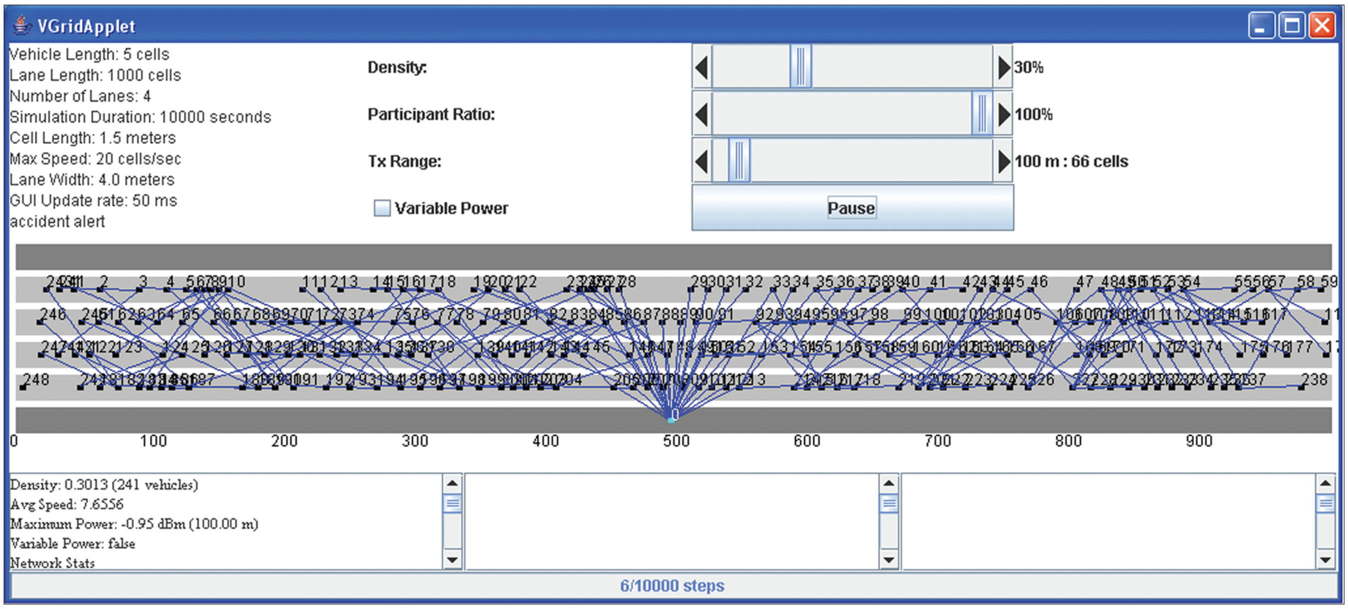


Figure 3. *A snapshot of the graphical user interface of VGSim. Lines connecting vehicles show communication links between vehicles.*

vehicles. SWANS also includes standard 802.11 medium access control (MAC) layer protocol and several path loss and fading models at the physical layer.

The introduction of close-loop interaction between the two simulation modules is achieved by injecting driver decision process into different applications. At each time step, each driver/vehicle makes the decision on how to change the speed/position of the vehicle according to not only traffic conditions perceived, but also the traffic control messages received from the wireless channel. Figure 4 shows the interactions between the major VGSim components during simulation. The `RoadEntity` object maintains the main simulation loop by providing an implementation of the `run()` method of the `Proxiable` interface in JiST/SWANS, which makes the `RoadEntity` a simulation entity thread in JiST/SWANS. The `run()` method and the `moveVehicle()` method of each vehicle object are invoked at each time step. Upon invocation, each vehicle object calls the mobility model object's `updatePos()` method to get an updated position information according to the mobility model logic. Then the updated position information is fed into the application's update method `updateApp()`. This method implements the logic of the wireless communication network and traffic control applications. At the end of each time step, each vehicle updates its own properties such as the position for the next time step, speed limit, probability of acceleration or deceleration, and the probability of lane change according to updated application state. These updated properties will result in changes in the behavior of vehicle movement in the next time steps. In the case shown in Fig. 4, the mobility model is an implementation of the N-S model, and the Variable Speed Limit (VSL) application is installed on the vehicle.

Mobility Model: Validation and Extension

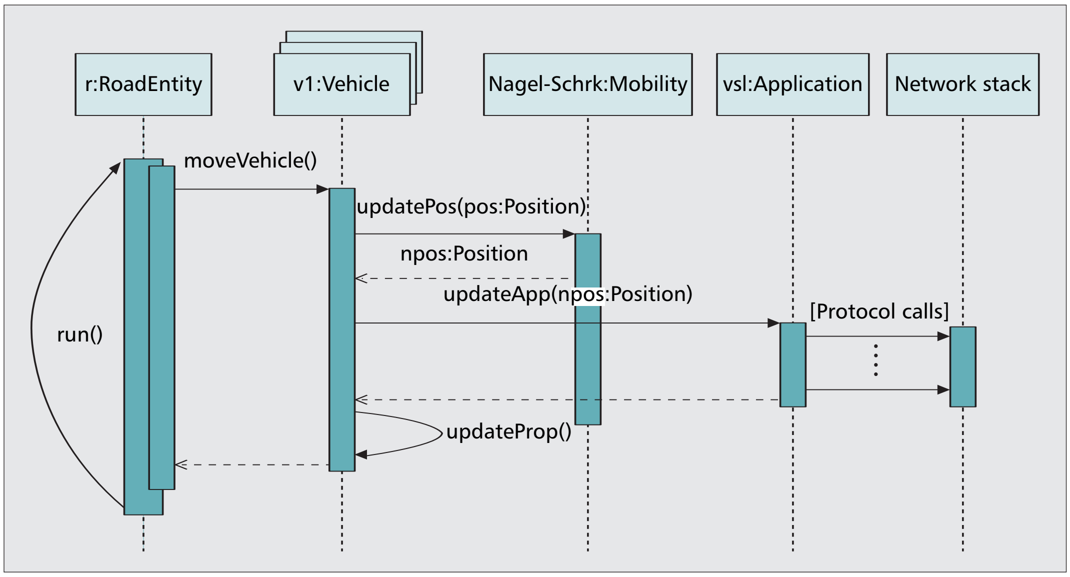
As a vital part of VGSim, the mobility model's accuracy determines the overall accuracy of the simulation. In this section we first describe the modifications we made to the original N-S model and the validation. Then we describe how VGSim can be extended to accommodate other mobility models.

Validation of the Finer-Grained N-S Model

In VGSim we have adopted the classic N-S mobility model used extensively in vehicular traffic engineering research. The original N-S model's temporal-spatial resolution is adequate for vehicular traffic engineering research. However, for evaluating the performance of wireless communication, the temporal resolution in terms of seconds is too coarse-grained. Therefore, updating the N-S model with a finer resolution is necessary for accurate VANET simulation. However, merely changing the resolution in the original N-S model results in inaccurate vehicular traffic generation. Therefore, we modified the original N-S model, adding more realistic acceleration, deceleration, and lane changing behaviors. A detailed description of the modified finer-resolution N-S model is reported in [22].

In order to validate our refined fine-grained mobility model we compared the data obtained from our model with real world traffic data. For the latter, we used the vehicle traces produced by the NGSIM project [23]. Ideally, the more accurate the mobility model, the higher the degree of correlation with the NGSIM data.

Our simulation setup consists of a five-lane 700-ft (213 m) highway. In order to be able to accurately compare with the NGSIM data, we must guarantee that the initial and road boundary conditions in our simulation are the same as those in the NGSIM data set [23]. The details of how we reproduce the initial and road boundary



■ **Figure 4.** *Interactions among the VGSim components.*

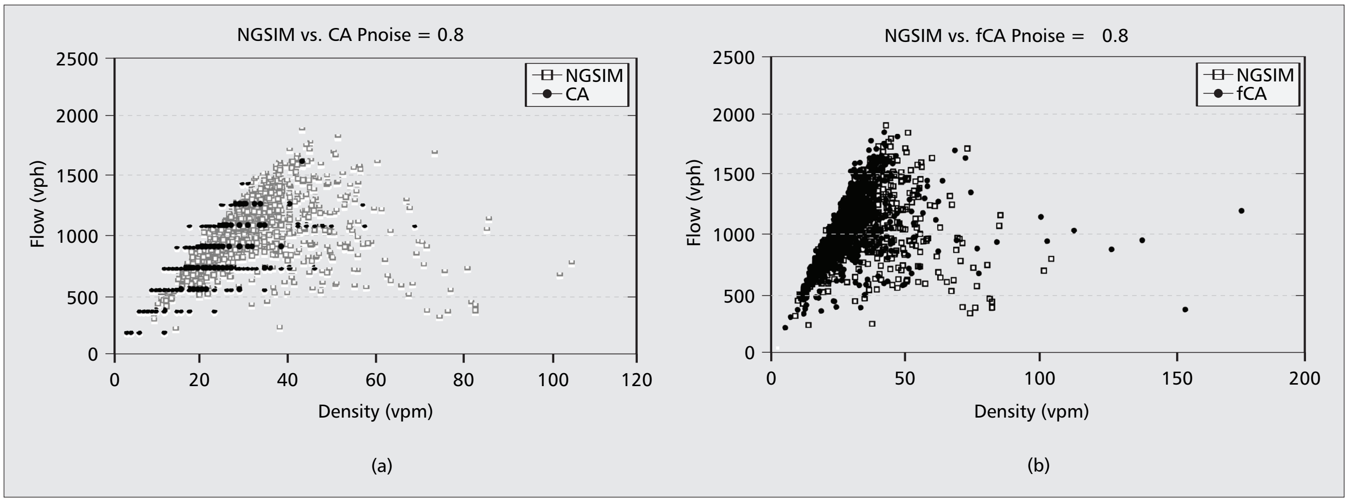
conditions in our simulation can be found in [22].

Our comparison is based on the fundamental diagram (flow-density diagram) [22]. In order to show the accuracy of our finer resolution mobility model, we first compare the fundamental diagram for the basic N-S models with that of NGSIM. It is known that N-S models can produce a triangular flow-density fundamental diagram. However, matching the fundamental diagram generated by the CA model with real traffic data is a challenging task. Figure 5a shows the fundamental diagram generated by the original N-S model (CA). It shows that in this case of random slowdown noise of 0.8, the CA model can generate the required triangular shaped diagram; however, it fails to match the NGSIM data set.

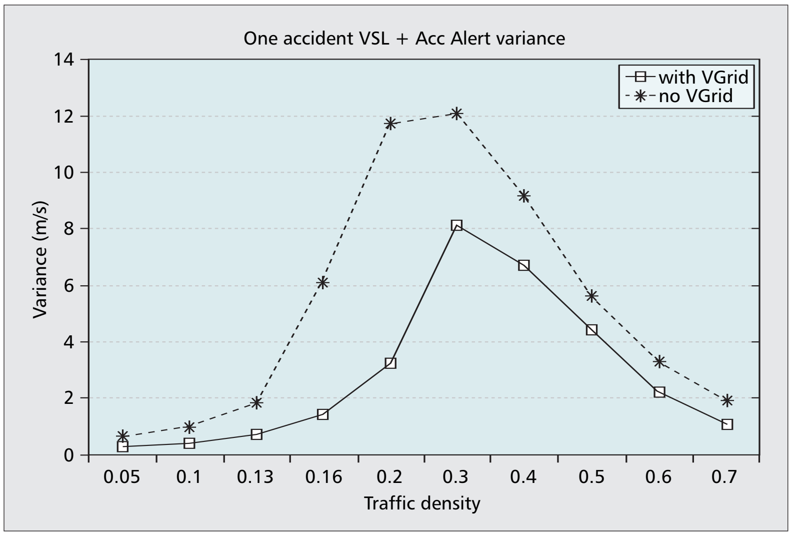
Figure 5b shows the fundamental diagram generated by our finer resolution model denoted fCA. This diagram shows that our finer resolution model not only reproduces the classic triangular flow-density diagram, but also matches with the real traffic data from NSGSIM better than the original N-S model. This guarantees the accuracy of VANET simulation at higher spatial and temporal resolution.

Extending VGSim to Other Mobility Models

As discussed in the previous section, commonly adopted mobility models in vehicular traffic engineering may not completely fulfill the requirements for VANET research. Therefore, modification of common mobility models or even incorporating a totally different mobility model for better VANET simulation may be required. Because of VGSim's modular design, this is easily achieved by providing an implementation for a mobility model interface in Java. The mobility model interface in VGSim only has one method that must be implemented (`updatePos()` as shown in Fig. 4). The `updatePos()` method contains logic of how to update vehicle positions in any time step. There-



■ **Figure 5.** *Comparison of the fundamental diagrams obtained using different mobility models and NGSIM data set: a) original N-S model (CA) vs. NGSIM; b) finer resolution N-S model (fCA) v.s. NGSIM*



■ **Figure 6.** *Average speed variance with one accident (w/ VGrid implies all vehicles implement the Acc Alert and VSL applications).*

fore, extending VGSim using other mobility models simply takes two steps:

1. Implementing a mobility class with `updatePos()` method
2. Associating each vehicle with an object of the mobility class

After the simulation is executed, the `updatePos()` is invoked every time step for each vehicle. All other tasks, including placing vehicles on the road, ensuring that there are no collisions, performing wireless communication simulation, and visualization, do not require any modification in VGSim. Although VGSim is currently using an in-house implementation of the mobility model, it is also possible for VGSim to adopt other standalone microscopic traffic simulators. This is achieved by implementing an interface wrapper for controlling and/or communicating with the simulator. The ability of extending VGSim to other mobility models ensures that VGSim is not tied to one specific mobility model or vehicular traffic simulation platform. This is a limitation in many other VANET simulators that integrate with standalone traffic simulators.

VGSim Application Development and Examples

VGSim Application Development

Another advantage of VGSim is the ease with which VANET applications can be developed. Due to JiST/SWANS's flexibility of performing wireless simulation, embedding wireless simulation in the rest of the application logic is simply achieved by providing a Java class with an implementation of the `updateApp()` method (Fig. 4). Therefore, it is possible to have multiple applications executing in the vehicles simultaneously. In fact, in the current VGSim, multiple applications can be turned on at the same time. Some applications provide basic services such as position beaconing. Other more complex applications can make use of the service provided by other applications such as location-aware ad hoc routing.

Application Example: Accident Alert and Optimal Variable Speed Limit

To demonstrate the capability of our simulator, we built two VANET traffic control applications: Accident Alert (Acc Alert) and VSL on top of VGSim. The Accident Alert application utilizes the vehicle's onboard wireless communications to send alerts to upstream vehicles of the presence of an obstruction in the road ahead. This will allow them to change out of impacted lanes earlier and also prevent them from changing into those lanes. For VSL, vehicles acquire position and velocity information of other vehicles through wireless communication and then cooperatively compute the appropriate speed limit for different sections of the road. Both applications are intended to smooth vehicular traffic on highways. Figure 6 shows the average speed variance with and without the VGSim supporting Acc Alert and VSL application, with one accident simulated on the road. We can see a significant decrease in variance with the use of VSL and Acc Alert. Both Acc Alert and VSL applications are vehicular driver safety and traffic control applications. Without VGSim's support of closed-loop interaction between the network simulation and the microscopic vehicle traffic simulation, it is hard to evaluate the effectiveness of both applications.

Conclusion

Simulation is one of the most commonly used tools in VANET studies. In this article we first discuss the classification of simulation tools for VANET applications and the architectural requirements for accurate simulations. After presenting a review of simulation tools used in VANET research, we present VGSim, which can fulfill most requirements of accurate simulation. It implements closed-loop integration of realistic vehicular traffic and a wireless communication simulation module. It is highly flexible and can easily adopt different mobility models. The application development process is easy and suitable for building multiple distributed VANET applications that can execute concurrently. Additionally, since it executes as a standalone Java application using the efficient JiST/SWANS package, it is more resource efficient than approaches that integrate existing network and traffic simulators. We validate the accuracy of the mobility model of our simulator. Finally, we present results of Accident Alert and VSL as proof-of-concept applications simulated using VGSim.

References

[1] Network Simulator 2; http://nsnam.isi.edu/nsnam/index.php/user_information
[2] QualNet; http://www.scalable-networks.com/products
[3] Jist/SWANS; http://jist.ece.cornell.edu
[4] D. Choffnes and F. Bustamante, "An Integrated Mobility and Traffic Model for Vehicular Wireless Networks," *Proc. ACM VANET '05*, Sept. 2005.
[5] OMNet++; http://www.omnetpp.org
[6] VISSIM; http://www.english.ptv.de/cgi-bin/index.pl
[7] CORSIM; http://mctrans.ce.ufl.edu/featured/TSIS/Version5/corsim.htm

[8] SUMO; http://sumo.sourceforge.net/
[9] A. K. Saha and D. B. Johnson, "Modeling Mobility for Vehicular Ad Hoc Networks," *Proc. ACM VANET '04*, Aug. 2004.
[10] Y. Zhang, J. Zhao, and G. Cao, "On Scheduling Vehicle- Roadside Data Access," *Proc. ACM VANET '07*, July 2007.
[11] M. Caliskan, D. Graupner, and M. Mauve, "Decentralized Discovery of Free Parking Places," *ACM VANET '06*, July 2006.
[12] J. Yin *et al*., "Performance Evaluation of Safety Applications over DSRC Vehicular Ad Hoc Networks," *Proc. ACM VANET '04*, Aug. 2004.
[13] H.-Y. Huang *et al*., "Performance Evaluation of SUVnet With Real-Time Traffic Data.," *Proc. IEEE Trans. Vehic. Tech*., vol. 56, no. 6, July 2007, pp. 3381–56.
[14] D. Li *et al*.," A Distance-Based Directional Broadcast Protocol for Urban Vehicular Ad Hoc Network," *Proc. Int'l. Conf. Wireless Commun., Networking, and Mobile Comp. 2007*, Sept. 21–25, 2007, pp. 1520–23.
[15] M. Piorkowski *et al*., "Joint Traffic and Network Simulator for VANETs" *MICS 2006*, Zurich, Switzerland, Oct. 2006.
[16] C. Lochert *et al*., "Multiple simulator interlinking environment for IVC," *Proc. ACM VANET '05*, Sept. 2005.
[17] C. Sommer and F. Dressler, "The DYMO Routing Protocol in VANET Scenarios," *Proc. 66th IEEE VTC 2007-Fall*, Baltimore, Maryland, Sept./Oct. 2007, pp. 16–20.
[18] C. Sommer *et al*., "On the Need for Bidirectional Coupling of Road Traffic Microsimulation and Network Simulation," *Proc. 1st ACM Int'l. Wksp. Mobility Models Networking Research*, May 2008, pp. 41–48.
[19] S. Y. Wang *et al*., "NCTUns 4.0: An Integrated Simulation Platform for Vehicular Traffic, Communication, and Network Researches," *Proc. 1st IEEE Int'l. Symp. Wireless Vehic. Commun.*, Baltimore, MD, Oct. 2007.
[20] C. Gorgorin *et al*., "An Integrated Vehicular and Network Simulator for Vehicular Ad-Hoc Networks," *Proc. 20th Euro. Simulation Modeling Conf*., Oct. 2006.
[21] K. Nagel and M. Schreckenberg, "A Cellular Automaton Model for Freeway Traffic," *J. Physique*, vol. 2, 1992, pp. 2221–29.
[22] M. Zhang and H. Du, "Finer-Resolution Cellular Automata Model for Intervehicle Communication Applications," *87th Annual Meeting Transportation Research Board*, 2007.
[23] NGSIM; http://ngsim.fhwa.dot.gov

BIOGRAPHIES

BOJIN LIU (frdliu@ucdavis.edu) is a Ph.D. student in the Computer Science Department, University of California, Davis. He received his Bachelor degree in computing from Hong Kong Polytechnic University. His research interests include vehicular ad hoc networks, wireless networks, and parallel and distributed systems.

BEHROOZ KHORASHADI (bkhorashadi@ucdavis.edu) is a Ph.D. student in the Department of Computer Science at the University of California, Davis. He graduated from the University of California, Berkeley in May 2004. He is currently working in the Networks laboratory with a focus on peer-to-peer networks and VGrid (vehicular ad-hoc Networks). VGrid is a vehicular ad hoc networking and computing grid for intelligent traffic monitoring and control. The goal is to evolve the intelligent transportation system (ITS) from a centralized to a distributed approach, in which vehicles can cooperatively solve traffic-flow control problems autonomously. One example application is the lane-merging scenario, especially when visibility is low during storms or foggy weather.

HAINING DU (hndu@ucdavis.edu) is a Ph.D. candidate in the Civil and Environmental Engineering Department at the University of California, Davis. His dissertation research attempts to develop improved microscopic vehicular traffic models for distributed traffic control via DSRC enabled vehicles.

DIPAK GHOSAL (dghosal@ucdavis.edu) received a B.Tech. degree in electrical engineering from the Indian Institute of Technology, Kanpur in 1983 and an M.S. degree in computer science and automation from the Indian Institute of Science, Bangalore in 1985. He received his Ph.D. degree in computer science from the University of Louisiana in 1988. He is currently a professor in the Department of Computer Science at the University of California, Davis. His primary research interests are in the areas of high-speed and wireless networks with particular emphasis on the impact of new technologies on network and higher layer protocols and applications. He is also interested in the application of parallel architectures for protocol processing in high-speed networks and the application of distributed computing principles in the design of next generation network architectures and server technologies.

CHEN-NEE CHUAH (chuah@ucdavis.edu) is an associate professor in the Electrical and Computer Engineering Department at the University of California, Davis. She received her B.S.E.E. from Rutgers University, and her M. S. and Ph.D. in electrical engineering and computer sciences from the University of California, Berkeley. Her research interests include Internet measurements, network management, and wireless/mobile computing. She is an Associate Editor for *IEEE/ACM Transactions on Networking*.

MICHAEL ZHANG (hmzhang@ucdavis.edu) is an associate professor in the Electrical and Computer Engineering Department at the University of California, Davis. She received her B.S.E.E. from Rutgers University, and her M. S. and Ph.D. in eectrical engineering and computer sciences from the University of California, Berkeley. Her research interests include Internet measurements, network management, and wireless/mobile computing. She is an Associate Editor for *IEEE/ACM Transactions on Networking*.