

Comparative Study of Connected Vehicle Simulators

Md Salman Ahmed, Mohammad Asadul Hoque, Phil Pfeiffer

Department of Computing

East Tennessee State University

ahmedm@goldmail.etsu.edu, hoquem@etsu.edu, phil@etsu.edu

Abstract—Contemporary studies of Intelligent Transportation Systems (ITSes) use simulations of vehicular and communications traffic, due to the ethical and practical infeasibility of conducting experiments on real transportation networks. Different simulators have been developed for modeling real-time vehicular mobility and inter-vehicular communication under varying traffic and roadway conditions. While most model the effect of mobility on communications, only a few simulate the impact of inter-vehicular communication on vehicular mobility. None, moreover, are implemented as parallel or distributed frameworks: an essential requirement for the study of ITS applications in large-scale urban environments. As a starting point for developing such a framework, one contemporary simulator, VNetInetSim, was tested to determine its behavior under large loads. Testing determined that VNetInetSim’s memory usage and execution time increase exponentially in the number of simulated vehicles while remaining relatively constant under increased communication traffic.

Keywords—Intelligent Transportation System (ITS), Inter-vehicle Communication, Simulator, Vehicle dynamics, Vehicular Ad Hoc Network.

I. INTRODUCTION

Over the past few decades, a substantial increase in automobile usage has led to increases in highway congestion, incidents, fatalities and greenhouse gas emissions. In 2012 USA TODAY reported that Americans annually waste 1.9 billion gallons of gasoline in traffic on congested roads and pay more than \$100 billion in wasted fuel and lost time [1]. These adverse effects of automobile usage impact peoples’ lives and degrade the quality of the Earth’s environment.

Currently, automakers and technology developers like Google, Ford, and General Motors are making concerted efforts to improve surface transportation through Automated Vehicle (AV) technology [2], [3]. While AV can potentially reduce the stress of navigating traffic, its focus in most of cases is limited to the operation of vehicles in isolation from one another. This limitation is addressed by Connected Vehicle (CV) technology, which seeks to apply inter-vehicular communication to the development of safe, driver-friendly, and energy efficient assistive technologies for vehicle operation. One of the primary goals of CV research is the optimization of traffic flow across an entire transportation network through the exchange of information obtained through vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications. This exchanging of information, collectively known as V2X communications, could assist drivers in avoiding congestion, reducing vehicle stops, choosing a best route, and optimizing fuel efficiency.

The large scale deployment of CV technologies faces several challenges, particularly for urban environments. Evaluating the performance of CV-based safety-critical real-time applications in large-scale urban environments under varying traffic and roadway conditions is difficult, since these conditions can’t be generated in practice. Additionally, failures of CV-based applications may result in loss of lives. These issues can be addressed by using simulations to study and test ITS applications. Simulating ITS and CV systems, however, requires the integration and synchronization of two tightly coupled domains. The one, the transportation domain, models vehicular mobility, including traffic routing, car-following, lane-changing, vehicle dynamics, driver behavior, and traffic signal controls. The other, the communication domain, models mechanisms for data-traffic-related communications, including packet routing, end-to-end message delivery, and V2X-related cross-layer protocols. These two domains directly affect each other’s operation. For example, high speed traffic networks with high vehicle density may delay V2X communications and degrade communication quality [4]. On the other hand, communication delay and data loss may degrade the modeling of vehicular operation. Such degradations, even if minute, could adversely affect the ability of V2X-based applications to assure their users’ safety.

Efforts to develop a complete transportation simulator with a wireless network simulator for modeling and evaluating V2X-based ITS applications have been ongoing for the past decade. Older simulators fed fixed mobility trajectories to a communication network simulator. Many researchers [5]–[8] have studied the various mobility models developed for state-of-the-art simulators. However, a comparative modular analysis of different simulator components has yet to be written. Our current research, which focuses on the capabilities and limitations of existing sequential simulators in terms of their modular organization and architecture, has identified the need for a parallel simulation platform to support large-scale simulations of urban surface transportation systems [9].

The rest of the paper is organized as follows. Section II surveys the state of the art in CV simulators. Section III summarizes this survey’s findings in tabular form. Section IV presents the results of preliminary load tests of VNetInetSim, a contemporary ITS simulator, and what they reveal about the simulator’s scalability. Section V concludes with considerations related to the implementation of parallel simulators for evaluating large scale urban vehicular networks.

II. STATE OF THE ART VANET SIMULATORS

Current Vehicular Ad Hoc Network (VANET) simulators can simulate the impact of vehicular communication on transportation systems. Some simulators can also create dynamic

mobility trajectory traces and mobility models. Examples of these simulators include ASH, STRAW, Veins, VnetIntSim, TraNS, iTETRIS, GrooveSim, and Automesh.

A. ASH

Application-aware SWANS with Highway mobility (ASH) [10] provides an application-aware mobility model using two-way communication between a vehicular mobility model and a network simulator. Ibrahim and Weigle use the term “application-aware” to emphasize ASH’s simulation of safety considerations such as alert information and lane-changing through two-way communication.

ASH extends work by various authors. Its supporting modules include the Scalable Wireless Ad hoc Network Simulator (SWANS) [11], which ASH uses as its network model; the Intelligent Driver Model (IDM) [12] module, which models how cars follow other cars; the Minimizing Overall Braking decelerations Induced by Lane changes (MOBIL) [13] module, which uses an incentive criterion for lane attractiveness and a safety criterion to model lane changes; and a node model for its mobility model. ASH also uses the Inter-Vehicle Geocast (IVG) [14] and probabilistic IVG (p-IVG) [15] protocols to broadcast messages.

ASH extensions to SWANS include the following:

- *Modeling two-way communication between the mobility and networking models.* ASH implements two-way communication by using its application layer to override IDM/MOBIL’s normal behavior through acceleration, deceleration, and lane-change mobility control primitives.
- *Modeling highway topology.* ASH’s configuration file specifies road segment characteristics such as segment length, number of directions, number of lanes, and the number and locations of exits and entries.
- *Modeling mobility states.* ASH’s node model represents a participating vehicle as a mobile communicating node, a non-participating vehicle as a mobile silent node, a roadside unit as a static communicating node, and a road obstacle as a static silent node. Participating vehicles run user-defined applications at simulation time whereas non-participating vehicles run a null application.
- *Intelligent broadcast.* In place of flooding-based broadcasting, ASH uses the IVG algorithm with a timer for node broadcast. IVG reduces network traffic by using a timer to expire broadcast messages.
- *Logging and statistical facilities.* ASH supports logging utilities at different levels including the simulation, lane, vehicle, and message type levels. It also maintains the statistical simulation data of every vehicle in order to answer statistical queries.

B. OVNIS

Pigne et al. describe OVNIS as a realistic vehicular network management platform that can adjust node mobility and generate vehicular traces at runtime [16]. OVNIS manages

an interconnection between the Simulation of Urban Mobility (SUMO) traffic simulator [17], a vehicular mobility simulator that supports programmed interaction through Application Program Interfaces (APIs), and network simulator 3 (ns-3) [18], a wireless network simulator that can simulate about 20000 nodes in a network. OVNIS also embeds a tool that generates vehicular traces based on real traffic data.

OVNIS’s Traffic Aware Network Manager, the network management platform’s main component, maintains a feedback-based interconnection with its traffic simulator and nodes applications modules. The Traffic Aware Network Manager module does the following during simulation:

- Starts, initializes and operates the network simulator.
- Starts the traffic simulator.
- Allows the nodes applications module to query the traffic simulator about every node’s speed, position, speed limit, and lane number.
- Iteratively pulls mobility information from the traffic simulator.
- Manages node mobility according to the pulled mobility information.

Pigne et al. evaluated OVNIS using two experiments. The first tested OVNIS’s overall computation performance based on its radio signal ranges. The experimental data shows that “the smaller the range, the faster the computation.” The second experiment evaluated OVNIS’s correctness, based on the extent to which simulated vehicles changed routes as the volume of vehicles increased. Their experiments showed that the vehicles’ average speed decreases and inter-vehicular communication increases with an increase in the volume of vehicles. Then the vehicles start finding alternative routes and managing their routes.

C. STRAW

Choffnes and Bustamante’s STreet RAndom Waypoint (STRAW) [19] application supports the modeling of vehicular motion in urban roads. STRAW can model road segments, intersections, traffic control mechanisms, and individual vehicles, including high speed vehicles and inter-vehicular communication. STRAW’s support for modeling individual vehicles, according to its authors, distinguishes it from earlier VANET simulators.

STRAW treats a vehicle as a node with a set of properties, including maximum speed, reaction time and acceleration rate. Road segments, or portions of roads between two intersections, are modeled according to their shape, length, width, name, speed limit, class and address attributes. Traffic control mechanisms provide deterministic admission control protocols for vehicles at each intersection.

STRAW is architected as a system of three interacting component models. They include an intra-segment mobility model, an inter-segment mobility model, and a route management and execution model.

The intra-segment mobility model simulates vehicle motion within individual road segments. Motion is simulated using a

car-following mechanism that accounts for the speed of the vehicle that a simulated vehicle is following and the distance to that vehicle. Vehicles use this model to accelerate to a maximum limit and decelerate on encountering speed limits, stop signs and stoplights.

The inter-segment mobility model determines how vehicles behave at intersections. The model applies a deterministic admission control protocol to determine how vehicles accelerate and decelerate. It also determines a vehicle's waiting time at stop signs and stop lights.

The route management and execution model determines the road segment that a vehicle will enter when it crosses an intersection. The model can choose this segment using a deterministic or a stochastic strategy. The deterministic strategy selects the next segment based on the fastest time and shortest distance to a preassigned destination, as calculated by the A* search algorithm. The stochastic strategy assigns probable road choices to a vehicle based on its trajectory. It then uses a probability value at each intersection to select the next segment.

STRAW supports two strategies for modeling driver response to vehicular collisions. In the particle system approach, a vehicle detects and reacts to collision events. In the vehicular approach, a vehicle detects collisions and avoids them when it can.

According to Choffnes and Bustamante, STRAW's mobility model is general enough to integrate into any wireless network simulator. The model performs well in terms of memory usage, but the computation cost is high for large numbers of vehicles. The model also fails to support the dynamic allocation and deallocation of vehicle nodes and lane changing.

D. Veins

The Vehicles in Network Simulation (Veins) [20] is a hybrid framework for evaluating the impact of inter-vehicular communication (IVC) protocols on road traffic mobility. Veins consists of a network simulator, a road traffic simulator, and a communication channel that supports the active exchange of control and data between the two simulators.

Veins' network simulator, OMNeT++ [21], is an event based simulator that simulates VANET protocols with the help of Veins' INET Framework extension. OMNeT++ represents VANET scenarios as hierarchical modules and stores the relationship and communication links between modules in network description files. Connectivity protocols such as TCP, UDP, IPv4, and ARP are added to OMNeT++ as extensions by the INET Framework.

Veins' road traffic simulator extends SUMO with Krauß's (1998) car-following mobility model. According to Sommer et al. [22], combining SUMO with the IVC protocols provides better simulation results than SUMO alone.

Veins uses dedicated modules to support bidirectional communications between OMNeT++ and SUMO. These modules use a TCP connection to exchange simulation commands and mobility traces. Each simulator buffers commands as it receives them and processes commands in the order received.

Commands are processed in rounds, as follows. At each time step, OMNeT++ sends all buffered commands to SUMO. SUMO simulates a round of traffic, then replies with a series of commands and generated mobility traces. OMNeT++ uses the traces to reconfigure the movement of nodes (vehicles). OMNeT++ allows nodes to alter their speeds and routes according to IVC, if all commands are processed and nodes reconfigured before next scheduled time step.

Sommer et al. used Veins to evaluate the impact of two IVC protocols on VANET scenarios. In the one protocol, vehicles communicate directly to a dedicated centralized Traffic Information System (TIS) using TCP connections and standard MANET (Mobile Ad Hoc Network) protocols. Vehicles exchange incident warnings with the TIS at intervals of 60s or 180s depending on road traffic. The TIS also maintains connections with roadside units in order to improve IVC. In the other protocol, vehicles maintain inter-vehicle communications by distributed or self-organized TIS using UDP broadcast communication. Incident warnings are flooded through VANET by UDP broadcast. When a vehicle gets a warning message, it queries the originating vehicle to determine if the warning is current.

The authors evaluated the protocols' impacts on vehicular mobility using a Manhattan grid and a real street map. In both cases, the authors ran four sets of simulations:

- One where vehicles were free to move without any interruption, with no IVC.
- One where the leading vehicle was stopped for a short duration with no IVC.
- Two where the vehicles' average speeds were calculated based on small and large scale simulations with the support of IVC. The small scale and large scale simulations used 5 hops and 25 hops to disseminate information, respectively.

Stationary vehicles in these experiments reported incidents using timestamped warning messages. Upon identifying these incidents, the network simulator stored the incident information and adjusted travel time for the stationary vehicles. The simulation then used Dijkstra's shortest path algorithm to compute new routes that bypassed the incident for the segment's other vehicles.

In both sets of experiments, the average speed of the first, third and fourth runs was greater than the second run. This indicates that stopping the leading vehicle in the second set of simulations caused congestion that increases other vehicles' travel time. During the third and fourth runs, those vehicles used inter-vehicle communication to get congestion information, then change their routes and increase their average speed.

E. VNetIntSim

Vehicular Network Integrated Simulator (VNetIntSim) [23] provides a modeling and simulation framework for VANETs and Intelligent Transportation System (ITS) applications. Vnet-IntSim consists of linker modules that integrate the INTEGRATION traffic simulator [24] with the OPNET communication network simulator [25]. These modules provide a two-way communication channel between INTEGRATION and OPNET.

Four modules drive VnetIntSim's operation. VnetIntSim's configuration reader module specifies an XML topology file containing vehicle specifications for configuring OPNET. VnetIntSim's communication module creates a shared memory region for the INTEGRATION and OPNET simulators, which then exchange information through shared memory. INTEGRATION's location module calculates vehicular locations and sends them to OPNET's driver module. Finally, its driver module checks simulation time from the received information, identifies simulation time mismatches, fixes inconsistencies and updates the vehicles' information.

When VnetIntSim starts execution, it establishes a communication channel between INTEGRATION and OPNET. First, the two simulators exchange hello messages to create the connection. The simulators then synchronize their simulation attributes, interval, and duration; the number of vehicles; and network size.

After successful synchronization, VnetIntSim enters its simulation loop. The VnetIntSim simulator primarily does movement-based simulation. It provides updates on the number of moving vehicles in a network, their locations, and traffic density. Though the simulator can simulate simple vehicle-to-vehicle and vehicle-to-infrastructure scenarios consistently, it fails to simulate large-scale scenarios.

F. TraNS

The Traffic and Network Simulation Environment (TraNS) simulator [26] simulates VANETs, accounting for vehicular mobility. TraNS supports two modes of simulation. In network-centric simulation, TraNS simulates statically determined traffic flows (e.g. music or travel information) [27]. The traffic simulator generates a simulation trace and the network simulator simulates the trace file. In application-centric simulation, TraNS allows dynamically generated exceptional events (e.g. abrupt braking and collision avoidance) to alter traffic [28]. Since the traffic and network simulators can run concurrently in application-centric simulation, no trace file is generated. As a result, this approach reduces the memory consumption for large-scale simulation.

G. iTETRIS

The Integrated Wireless and Traffic Platform for Real-Time Road Traffic Management Solutions (iTETRIS) [29] simulates ITS applications on large-scale vehicular networks. iTETRIS supports WiMAX, UMTS, and DVB-H wireless and radio access technologies. iTETRIS is the first simulator to support the European Telecommunications Standard Institute (ETSI) ITS G5A standard.

According to Rondinone et al., iTETRIS achieves accurate simulations for realistic and complex traffic scenarios. Its modular architecture supports the integration of external modules. iTETRIS proper is a front-end for ns-3 and SUMO. It accepts input on roads and traffic in a SUMO-compatible format. The iTETRIS Controlling System interacts with SUMO and ns-3 and synchronizes simulation data with ITS applications using push-pull command mechanisms.

iTETRIS's accuracy for simulations of low- and mid-density traffic is better than its simulations of high-density

traffic. Its features include providing information on fuel consumption and traffic congestion along with suggesting speed and route changes accordingly.

H. GrooveSim

GrooveSim [30] simulates inter-vehicular communication and vehicular mobility in a road traffic network using the authors' communication and mobility model and the GrooveNet routing protocol. GrooveNet, a hop-based communication protocol, uses a dedicated short range communication based transceiver, a global positioning system, a cellular modem, and audio/video devices to broadcast data and information over multiple hops.

GrooveSim represents a vehicular network as a planar graph whose edges represent road segments and whose vertices represent intersections. Road segments are modeled using Topologically Integrated Geographic Encoding and Referencing (TIGER) [31] records that contain the segments' names, types, locations (latitude and longitude), addresses, and speed limits. The graph abstraction is used for the shortest path calculation and region partitioning.

GrooveSim supports an on-road driving mode, a virtual traffic network simulation mode, a playback mode, a hybrid simulation mode, and a test scenario generation mode. In its driving mode, a real vehicle sends warning messages to other real vehicles using the GrooveNet portable networking kit and sends warning messages. In simulation mode, GrooveSim simulates a virtual road traffic network based on vehicular mobility and communication models. In playback mode, it replays simulations of vehicular movement and communication using drive and simulation mode logs. In hybrid simulation mode, it simulates real and virtual vehicles on a road traffic network. In test generation mode, it generates parameterized simulation scenarios using models that include vehicles' IDs, speed models, origins, destinations, routes, and waypoints along the route.

GrooveSim defines its own mobility and communication models. The mobility model determines vehicular mobility based on a minimum and maximum speed, the number of vehicles on road segments, road segment speed limits, and a four-state Markov-based probabilistic model. The probabilistic model uses two states for city roads and two for highway roads. The communication model uses a two-state Gilbert-Elliott Markov model, a collision model, and a channel model to guarantee concurrent inter-vehicular communications.

GrooveNet's communication protocol uses a message diffusion mode to periodically exchange non-critical data such as congestion information. It uses a message directed mode to immediately exchange time-critical data such as alert messages. The protocol uses region based multi-hop routing in order to speed the communication and reduce message flooding overhead.

GrooveSim provides on-road crash warnings, sudden braking alerts, congestion information, traffic updates, and location based commercial services.

I. Automesh

The Automesh [32] simulation framework for ITS applications integrates five modules with three plug-in modules, as

follows:

- *Driving simulator module.* Automesh generates a dynamic mobility model for individual vehicles using an environmental model that supports speed limits and traffic signals. Automesh also accounts for vehicle dynamics including rates of acceleration and deceleration. This ability to dynamically generate mobility models distinguishes Automesh from other network and traffic simulators.
- *Network simulator module.* Automesh’s network simulator simulates inter-vehicle communication by using received data from the driving simulator’s dynamic mobility models to change driving behavior.
- *Propagation simulator module.* To evaluate the correctness and performance of communication protocols, Automesh provides a propagation simulator that simulates propagation calculation algorithms.
- *Geographic database server module.* This module provides geographic information such as road network information, a digital elevation model, and real 3D building information.
- *Automesh graphical user interface module.* This module provides a graphical user interface for configuring simulations and playing simulations’ animations.
- *Vehicle control plug-ins.* This module allows the driving simulator to attach custom driving behavior algorithms and custom mobility models to itself.
- *Propagation plug-ins.* This module allows custom wireless propagation models to interface to the network simulator.
- *Communication protocol plug-ins.* This module allows customized communication protocol stacks to interface to with network simulator.

III. COMPARATIVE SUMMARY

All of these simulators are implemented as sequential programs, though some could be modified to run in distributed and parallel computing environments. OVNIS, TranNS, GrooveSim, and Automesh model vehicular mobility dynamically using vehicle trajectory traces whereas ASH and STRAW use the car-following model. VnetIntSim and iTETRIS use linker modules to communicate between transportation and network modules whereas ASH, Veins and OVNIS use two-way communication. GrooveSim and Automesh also support the modeling of communication protocols. Table I summarizes these simulators’ names, their mobility models and their communication models.

IV. NEED FOR PARALLEL SIMULATION PLATFORM

Scalability is the most important limitation for all current platforms. VANETs and ITS simulation require high levels of scalability. Sequential simulations lack the processing resources to simulate urban transportation networks in real time. For example, in one experiment involving a sequential simulator [33], the simulation of a 200-node network created 4,600,000 events and required 16 minutes to process the events.

TABLE I: Summary of the above mentioned simulators

Simulators	Two simulation models of a simulator	
	Mobility model	Network model
ASH	IDM/MOBIL, IVG	SWANS
OVNIS	SUMO	NS-3
STRAW	Developed their own model	SWANS
Veins	SUMO, IVC	OMNET++
VnetIntSim	INTEGRATION	OPNET
TranNS	SUMO	NS-2
iTETRIS	SUMO	NS-3
GrooveSim	Developed their own model	Their own network model
Automesh	Customizable to add any mobility model	NS-2 or Qualnet

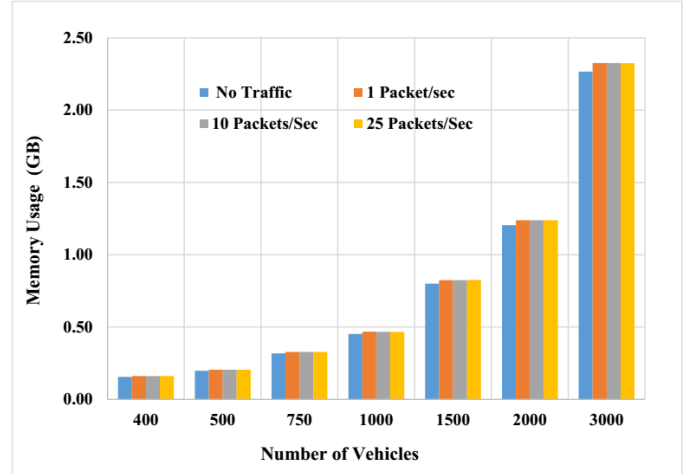


Fig. 1: The memory usage (GB) vs the number of nodes

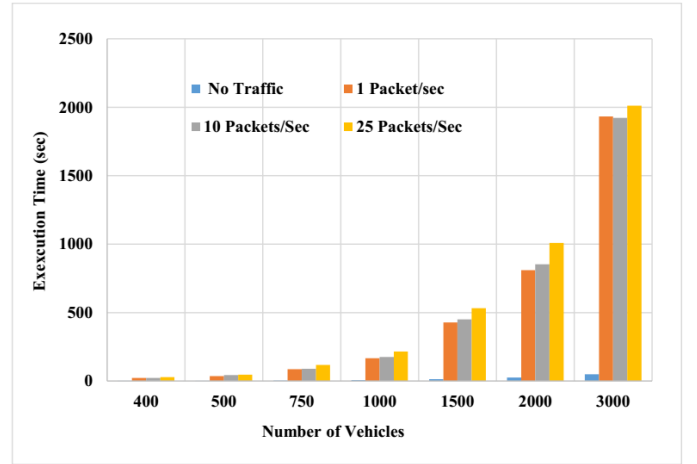


Fig. 2: The execution time (sec) vs the number of nodes

These levels of simulated traffic flow impose time, resource, and scalability constraints on sequential simulations of large-scale urban environments.

These observations motivated us to use VNetInetSim to analyze those factors that had the greatest impact on VANET scalability. We found that the number of wireless nodes (vehicles) and the data traffic rate per node were the primary

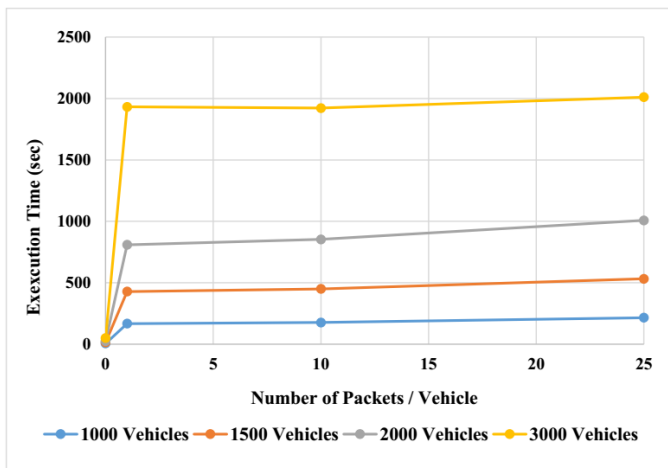


Fig. 3: The execution time (sec) vs traffic rate per vehicle

impediments to scalability. Our preliminary results show that memory usage and execution time increase exponentially with the number of vehicles in the system (Fig. 1 and 2). As shown in Fig. 1, increasing the data traffic rate for a given number of nodes has no significant effect on the memory usage. This is because OPNET, VNetInetSim's network simulator, discards packets when they reach their destinations, releasing their memory. These increases, however, do produce significant increases in simulation execution time (Fig. 2). This is to be expected. Fig. 3 shows a log-increase in the simulation time with respect to the traffic rate. These results were obtained on a machine of Intel Core-i7 Quad-core processor, 4 GB of memory, and running windows 7 Ultimate.

V. CONCLUSIONS

Most of the VANET simulators we surveyed can effectively simulate small-scale transportation networks. However, the simulation of large-scale urban environments will require parallel and distributed simulation. A parallel and distributed simulation platform must address the issues of optimal network partitioning, accurate parallel architecture, and synchronization between simulators. Graph-theoretical approaches and sparse matrix-based techniques could be used to achieve the necessary partitioning [34], while a parallel architecture that synchronizes separate communication and simulation modules could be used to structure this platform. We plan to investigate the challenges and issues pertaining to implementing parallel simulation platforms for the large-scale evaluation of CV-based urban transportation network.

REFERENCES

- [1] G. Stoller. (2007) Road congestion wastes 1.9 billion gallons of gas. Retrieved: 2015-11-5. [Online]. Available: <http://usatoday30.usatoday.com/money/industries/energy/story/2012-03-25/wasted-fuel-report/53776164/1>
- [2] Google. (2016) Google self-driving car project. Retrieved: 2015-02-09. [Online]. Available: <https://www.google.com/selfdrivingcar/>
- [3] B. CAREY. (2012) Shelley, stanford's robotic racecar, hits the track. Retrieved: 2016-02-09. [Online]. Available: <http://news.stanford.edu/news/2012/august/shelley-autonomous-car-081312.html>

- [4] M. Alam, M. Sher, and S. A. Husain, "Vanets mobility model entities and its impact," in *Emerging Technologies, 2008. ICET 2008. 4th International Conference on*. IEEE, 2008, pp. 132–137.
- [5] F. J. Martinez, C. K. Toh, J.-C. Cano, C. T. Calafate, and P. Manzoni, "A survey and comparative study of simulators for vehicular ad hoc networks (vanets)," *Wireless Communications and Mobile Computing*, vol. 11, no. 7, pp. 813–828, 2011.
- [6] S. A. Hussain and A. Saeed, "An analysis of simulators for vehicular ad hoc networks," *World Applied Sciences Journal*, vol. 23, no. 8, pp. 1044–1048, 2013.
- [7] M. K. Patel, "Comparative study of vehicular ad-hoc network mobility models and simulators," *International Journal of Computer Applications*, vol. 47, no. 6, pp. 38–43, 2012.
- [8] S. Khandelwal, "Comparative analysis of network simulator for vehicular adhoc networks (vanet) communication," *Journal of Advanced Computing and Communication Technologies*, vol. 2, no. 2, 2014.
- [9] M. A. Hoque, X. Hong, and B. Dixon, "Analysis of mobility patterns for urban taxi cabs," in *Computing, Networking and Communications (ICNC), 2012 International Conference on*. IEEE, 2012, pp. 756–760.
- [10] K. Ibrahim and M. C. Weigle, "Ash: Application-aware swans with highway mobility," in *INFOCOM Workshops 2008, IEEE*. IEEE, 2008, pp. 1–6.
- [11] R. Barr, "Swans-scalable wireless ad hoc network simulator," *March, URL; http://jist.ece.cornell.edu/docs.html*, 2004.
- [12] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical Review E*, vol. 62, no. 2, p. 1805, 2000.
- [13] A. Kesting, M. Treiber, and D. Helbing, "General lane-changing model mobil for car-following models," *Transportation Research Record: Journal of the Transportation Research Board*, 2007.
- [14] A. Bachir and A. Benslimane, "A multicast protocol in ad hoc networks inter-vehicle geocast," in *Vehicular Technology Conference, 2003. VTC 2003-Spring. The 57th IEEE Semiannual*, vol. 4. IEEE, 2003, pp. 2456–2460.
- [15] K. Ibrahim and M. C. Weigle, "Cascade: Cluster-based accurate syntactic compression of aggregated data in vanets," in *GLOBECOM Workshops, 2008 IEEE*. IEEE, 2008, pp. 1–10.
- [16] Y. Pigné, G. Danoy, and P. Bouvry, "A platform for realistic online vehicular network management," in *GLOBECOM Workshops (GC Workshops), 2010 IEEE*. IEEE, 2010, pp. 595–599.
- [17] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "Sumo-simulation of urban mobility," in *The Third International Conference on Advances in System Simulation (SIMUL 2011), Barcelona, Spain*, 2011.
- [18] T. R. Henderson, M. Lacage, G. F. Riley, C. Dowell, and J. Kopena, "Network simulations with the ns-3 simulator," *SIGCOMM demonstration*, vol. 14, 2008.
- [19] D. R. Choffnes and F. E. Bustamante, "An integrated mobility and traffic model for vehicular wireless networks," in *Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks*. ACM, 2005, pp. 69–78.
- [20] C. Sommer, R. German, and F. Dressler, "Bidirectionally coupled network and road traffic simulation for improved ivc analysis," *Mobile Computing, IEEE Transactions on*, vol. 10, no. 1, pp. 3–15, 2011.
- [21] A. Varga *et al.*, "The omnet++ discrete event simulation system," in *Proceedings of the European simulation multicongress (ESM2001)*, vol. 9, no. S 185. sn, 2001, p. 65.
- [22] S. Krauß, "Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics," Ph.D. dissertation, Universitat zu Koln., 1998.
- [23] A. Elbery, H. Rakha, M. Y. ElNainay, and M. A. Hoque, "Vnetintsim: An integrated simulation platform to model transportation and communication networks."
- [24] M. Van Aerde, B. Hellinga, M. Baker, and H. Rakha, "Integration: An overview of traffic simulation features," *Transportation Research Records*, 1996.
- [25] J. Prokkola, "Opnet-network simulator," *URL http://www.telecomlab oulu.fi/kurssit/521365A_tietoliikennetekniikan_simuloinnit_ja_tyokalu/Opnet_esittely*, vol. 7, 2006.

- [26] M. Piorkowski, M. Raya, A. L. Lugo, P. Papadimitratos, M. Grossglauser, and J.-P. Hubaux, "Trans: realistic joint traffic and network simulator for vanets," *ACM SIGMOBILE mobile computing and communications review*, vol. 12, no. 1, pp. 31–33, 2008.
- [27] G. F. Riley and T. R. Henderson, "The ns-3 network simulator," in *Modeling and Tools for Network Simulation*. Springer, 2010, pp. 15–34.
- [28] S. Rampfl, "Network simulation and its limitations," in *Proceeding zum Seminar Future Internet (FI), Innovative Internet Technologien und Mobilkommunikation (ITM) und Autonomous Communication Networks (ACN)*, vol. 57, 2013.
- [29] M. Rondinone, J. Maneros, D. Krajzewicz, R. Bauza, P. Cataldi, F. Hrizi, J. Gozalvez, V. Kumar, M. Röckl, L. Lin *et al.*, "itetriz: a modular simulation platform for the large scale evaluation of cooperative its applications," *Simulation Modelling Practice and Theory*, vol. 34, pp. 99–125, 2013.
- [30] R. Mangharam, D. S. Weller, D. D. Stancil, R. Rajkumar, and J. S. Parikh, "Groovesim: a topography-accurate simulator for geographic routing in vehicular networks," in *Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks*. ACM, 2005, pp. 59–68.
- [31] U. S. C. Bureau. (2015) Us geological survey (usgs) topographic maps. Retrieved: 2015-10-25. [Online]. Available: <http://www.census.gov/geo/maps-data/data/tiger.html>
- [32] R. Vuyyuru, K. Oguchi, C. Collier, and E. Koch, "Automesh: Flexible simulation framework for vehicular communication," in *Mobile and Ubiquitous Systems: Networking & Services, 2006 Third Annual International Conference on*. IEEE, 2006, pp. 1–6.
- [33] H. Wu, R. M. Fujimoto, and G. Riley, "Experiences parallelizing a commercial network simulator," in *Simulation Conference, 2001. Proceedings of the Winter*, vol. 2. IEEE, 2001, pp. 1353–1360.
- [34] M. Hoque, X. Hong, and B. Dixon, "Efficient multi-hop connectivity analysis in urban vehicular networks," *Vehicular Communications*, vol. 1, no. 2, pp. 78–90, 2014.