

An integrated simulation environment for testing V2X protocols and applications

Choudhury, Apratim; Maszczyk, Tomasz; Li, Hong; Dauwels, Justin; Math, Chetan Belagal

2016

Choudhury, A., Maszczyk, T., Math, C. B., Li, H., & Dauwels, J. (2016). An integrated simulation environment for testing V2X protocols and applications. *Procedia Computer Science*, 80, 2042-2052. doi:10.1016/j.procs.2016.05.524

<https://hdl.handle.net/10356/89727>

<https://doi.org/10.1016/j.procs.2016.05.524>

© 2016 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Downloaded on 03 Apr 2024 17:28:41 SGT

An integrated simulation environment for testing V2X protocols and applications

Apratim Choudhury¹, Tomasz Maszczyk¹, Chetan B. Math^{2,3}, Hong Li³, and Justin Dauwels¹

¹ Nanyang Technological University, Singapore

achoudhury@ntu.edu.sg, tmaszczyk@ntu.edu.sg, jdauwels@ntu.edu.sg

² Eindhoven University of Technology, Eindhoven, Netherlands

C.Belagal.Math@tue.nl

³ NXP Semiconductors, Eindhoven Area, Netherlands

hong.r.li@nxp.com

Abstract

Implementation of Vehicle-to-everything (V2X) communication technologies, for traffic management, has been envisioned to have a plethora of far-reaching and useful consequences. However, before any hardware/software infrastructure can be developed and implemented, a thorough phase of testing is warranted. Since actual vehicles and traffic conditions cannot be physically re-constructed, it is imperative that accurate simulation tools exist in order to model pragmatic traffic scenarios and communication amongst the participating vehicles. In order to realize this need of simulating V2X technology, we have created an integrated simulation environment that combines three software packages, **VISSIM** (traffic modelling), **MATLAB** (traffic management applications) and **NS3** (communication network simulation). The combination of the simulators, has been carried out in a manner that allows on-line exchange of data amongst them. This enables one to visualize whether a traffic management algorithm creates the desired effect and also the efficacy of the communication protocol used. In order to test the simulator, we have modelled the Green Light Optimized Speed Advisory (GLOSA) application, whose objective is communication of the present traffic signal phase information to oncoming vehicles using a transmitting unit installed on the signal itself. This information will allow the vehicles to calculate the desired speeds necessary to cross the relevant intersection without stopping. Therefore, a “Green Wave” can be created for all vehicles without the need to coordinate traffic signal timers, which can be rather complex in a multiple intersection traffic corridor.

Keywords: Simulation, V2X, VISSIM, NS3, MATLAB, GLOSA

1 Introduction

The communication paradigm of Vehicle-2-X[31] has garnered a considerable amount of attention in the past decade for the purpose of improving traffic safety[12],[32] and efficiency[17],[30].

The creation of the IEEE 802.11p [10],[14] standard and dedication of the 5.9 GHz frequency bandwidth for supporting communications for ITS applications, has motivated many countries to explore the possibility of implementing this technology on a wide scale, one of them being Singapore. This has led to the inception of the NTU-NXP Smart Mobility Test Bed [4] which will offer researchers the ideal playground to test V2X technologies and applications. This project is supported by the Economic Development Board of Singapore [1] and is being carried out by Nanyang Technological University and NXP semiconductors [5] in a joint collaboration. One part of this test-bed project is to develop a simulator that is capable of modelling and testing the capability of V2X on areas beyond the test bed and application scenarios which cannot be easily replicated on it.

In order to simulate V2X operations, there does not exist any self-supporting simulation tool. However, it is possible to integrate a number of different simulation softwares, that when operated together may allow one to analyse the effects of implementing V2X technologies. Following are the three major components of an integrated V2X simulator:

- **Traffic Simulator:** A traffic simulator is required to create accurate urban mobility models and investigate real-world traffic problems.
- **Network Simulator:** A network simulator is needed to build dynamic topologies between moving nodes, recreate protocols corresponding to a Vehicular Ad-Hoc Network (VANET) and simulate communication of vehicles with road-side units and with each other
- **Application Simulator:** There are a number of applications that can be designed to alleviate traffic problems. These may be broadly classified into “information exchange or distribution” (e.g. music or travel information), “safety applications” (e.g., abrupt braking, collision avoidance, etc.) or “vehicle re-routing” (information on traffic congestions, road-blocks, accidents further down the road). These applications and the information that they convey need to be simulated with as much practical detail as possible.

In order to make the simulator as realistic as possible we chose the microscopic traffic simulator VISSIM [6], NS3 [3] for network simulation and MATLAB [2] for application simulation. The reason we chose VISSIM is because it is able to (1) precisely replicate the trajectories of any single vehicle in simulation; (2) simulate right and left hand side movements; (3) fulfil the advanced transit and traffic signal requirements; (4) produce the vehicular and infrastructure (e.g. signals) information at every simulation interval as little as 100 ms; (5) provide real-time data exchange with communication simulator and external programs; and (6) manipulate with infrastructure components in a traffic network (e.g. cars dynamics and routes), as response to stimuli obtained from application/communication simulator. Though, the traffic simulator SUMO is used a lot for Vehicular communication research, it does not possess the capability of simulating left-hand driving, such as in Singapore. We chose NS3 for network simulation since it contains model libraries for simulating the the Wireless Access for Vehicular Environment (WAVE) [28] system architecture and is one of the most trusted simulators for testing V2X protocols. MATLAB has been chosen for application simulation since the algorithms for most V2X applications being envisioned for Singapore, will undergo initial testing using MATLAB. The integrated simulator will allow testing of those algorithms on a realistic simulation platform, without having to convert it to a different programming language. The application that was chosen for the purpose of demonstrating the simulator is the Green Light Optimized Speed advisory (GLOSA)[16], [27] application. We decided to go with GLOSA since it has been tested earlier on other simulation platforms and has the necessary results that we can use to

compare with the results derived from our simulator, as the main objective of this work is to test the simulator's application testing capability. This work also serves as a foundation step for us which we will use to further expand the capability of the simulation tool to carry out larger and more ambitious simulation exercises involving multiple vehicles and both safety and non-safety applications.

2 Literature Survey

Specifically, there are two ways to go about simulating V2X operations as elucidated in [23]. These are classified as a *network-centric* approach and an *application-centric* approach. The network centric approach pertains to offline analysis of the communication network [8], [13], [15], [20] and application-centric approach helps in online analysis where the effect of a V2X application on traffic is studied as well. A review of those works which emphasize on an application-centric simulation environment was carried out before embarking on this project.

One of the most widely used simulator coupling environments available is **VsimRTI**[25],[24] which uses IEEE standard for modelling and simulation (M & S) High-Level Architecture concepts to combine multiple simulators. It has been used in several other research endeavours for integration of simulators. Some of the simulators coupled using VsimRTI are SUMO and JiST/SWANS [16], MATLAB CCMSim (Car2X Channel Model Simulator) and OMNET++ [24] and SUMO and OMNET++[7],[24]. However, the aim of this work is to use VISSIM as the traffic simulator for which an ambassador interface is not available in VsimRTI. Also, there was not enough information with regards to coupling MATLAB using VsimRTI as we intend to use it for application simulation and not the default VSimRTI App. In addition to VsimRTI, several other integrated simulators have been developed. For example, there is [19] which integrates **VISSIM** (Traffic Modeling), **MATLAB** (Application Simulation) and **NS2** (Network simulation) to simulate a VANET (Vehicular Ad-Hoc network). However, there is no information whether this platform has been upgraded to work with NS3, since NS3 is significantly different from NS2 in several aspects. Also, there exists the Traffic and Network Simulation Environment (TraNS) [23] which links **SUMO** and **NS2** and the Integrated Wireless and Traffic Platform for RealTime Road Traffic Management Solutions (iTetris) simulator [18], which combines **SUMO** and **NS3**. Development of TraNS has ceased and does not support newer versions of SUMO or NS3 while iTetris is not suitable for our purposes since they use SUMO. Apart from these environments there exists the Veins framework which is an integration platform that couples **SUMO** and the INET framework from **OMNeT++**[21] and [9] whose authors have combined their custom traffic simulator **CARISMA** with **NS2**. Furthermore, there also exists the NCTUns 4.0 environment [29]. The issue with this environment is that traffic/mobility simulator is highly integrated with the network simulator and any realistic road traffic situations using external traffic simulators cannot be easily realized and tested [23] and [26]. However this platform was extended by coupling with VISSIM [22], which also inculcated the applications-dedicated message sets, SAE J2735.

For evaluating the simulator performance, the results were compared with the GLOSA simulations carried out in [16] and [27]. The primary objective of the cited works were to check the effectiveness of GLOSA in improving traffic performance based on parameters such as fuel consumption and vehicle stopped delay. In [16], the simulations were carried out using VsimRTI which couple SUMO and JiST/SWANS and results on decrease in fuel consumption due to GLOSA were used for comparison. [27] uses VISSIM 5.30 for traffic simulation and the Car2X module of VISSIM for network simulations. The Car2X module is no longer a feature of the newer versions of VISSIM. The GLOSA application was coded in C++ and interfaced

with VISSIM using the VISSIM Component Object Module (COM). We compared the vehicle stopped delay data obtained using our simulator with the ones published in this paper.

3 Simulator Design

Fig 1 shows the basic simulator coupling idea. VISSIM and MATLAB communicate with each other via VISSIM's COM interface, which allows access to most attributes of the traffic simulation such as vehicle speeds, positions, signal phase information, etc. through MATLAB. Both VISSIM and MATLAB have been installed on a Windows OS, since VISSIM has been designed to be used only with Windows. On the other hand, NS3 has been developed for Linux. To couple VISSIM/MATLAB with NS3, we used a Linux virtual machine, installed NS3 on it and linked the host machine (with Windows) to the virtual machine via a virtual network. We then used sockets API for communication of data between MATLAB and NS3.

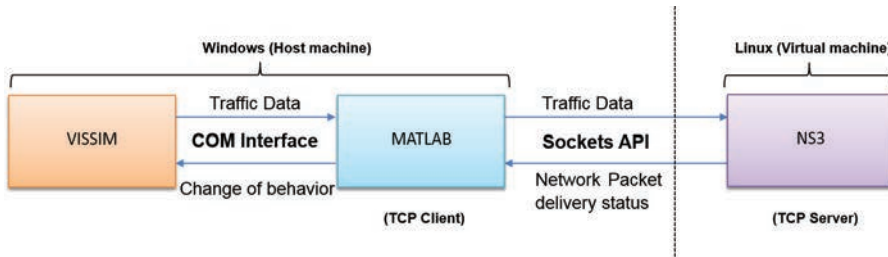


Figure 1: Block diagram of the simulation environment

In order to design an online simulator for V2X operations, one needs to take care of several considerations which are as follows:

- **Synchronization amongst simulators:** In order to make sure the results that are being produced by all the simulators are relevant for that time instant, the simulation times of the softwares used must be synchronized. VISSIM is controlled by MATLAB which updates the traffic model in a loop by executing the “RunSingleStep” command. RunSingleStep updates the simulations by 0.1 second (which is the resolution we chose for these simulations). On the other hand, NS3 is a discrete-event simulator which means that simulation time update is event based.
- **Dynamic addition and removal of nodes** It is evident that the simulation area and the VANETs that are being modelled will constantly see vehicles coming in and going out of the network. Also we intend to carry out simulations for considerable periods of time with variable traffic density, which means that the number of vehicles communicating every time instant may keep changing. This poses a challenge because NS3 can carry out simulations with only a constant number of pre-configured nodes. In order to carry out multiple simulations with variable number of nodes in NS3, in a single simulation run, a workaround has been proposed in subsection 3.2.
- **Selection of the appropriate mobility model in NS3** In order to make sure that the mobility information from VISSIM is correctly reflected in NS3, it is important that the appropriate mobility model is chosen. For this purpose we chose the inbuilt Waypoint Mobility Model. This model allows addition of waypoints, which prescribe the path the

vehicle is following, at any time. This allows the mobility model to be updated in an on-line manner. Every time two waypoints are provided to a node object in NS3, the simulations assume the vehicle to be moving between these two points with constant velocity. In the present simulations, waypoint information is acquired from VISSIM at each simulation step, which are then correspondingly updated in NS3.

3.1 Synchronization among simulators

In order to synchronize the three simulators, we utilized the blocking characteristics of sockets. First, the NS3 script goes through the configuration phase. Once it enters the simulation phase the code is blocked at the zeroth simulation second. Thereafter, the traffic simulation is initiated and is made to warm up. Once a steady state is reached, and the first set of vehicles enter the communication network their position information is sent to NS3 (queried from VISSIM by MATLAB) along with the time at which they enter the network. This allows scheduling a position update event in the network simulator at the same virtual time as VISSIM, which is currently blocked (through MATLAB) and is awaiting simulation results from NS3. Blocking event are scheduled every simulation second for obtaining data from MATLAB which ensures that at every subsequent second of VISSIM virtual time, appropriate vehicle data is communicated to NS3. A flowchart describing this process has been presented in Fig 2

3.2 Dynamic addition and removal of nodes

Since it is not possible to create and destroy nodes in NS3 once simulations have started, it was decided to create a fixed number of nodes, during the NS3 configuration phase (a number which is larger than the possible number of vehicle communicating in a particular scenario, at a given time) and not attach sockets. Sockets will be attached at runtime to as many vehicles as there are in the communication network. In order to characterize a vehicle in NS3, we created a Vehicle Node structure, which contains the NS3 node object and other information that is pertinent to the simulations and required for identifying and managing the nodes and the corresponding vehicle it represents. The C/C++ code is given below:

```
struct veh_Nodes
{
Ptr<Node> veh_node;    // A smart pointer to the NS3 node object
Ptr<Socket> veh_socket; // A smart pointer to the socket object attached to the node
int veh_number; // The vehicle number (obtained from VISSIM) associated with the node
bool in_use;    // true or false based on whether the node is being currently used
};
```

A vector of such vehicle structures objects are created and each NS3 node object is then loaded onto the `veh_node` member of the object. Once a new vehicle enters the communication network, its information will be loaded onto a Vehicle Node structure object currently not 'in use' and a socket will be attached. This step is carried out for every new vehicle that enters the network. Once a Vehicle Node is active, only its mobility model is updated every simulation second. To discontinue communication once the vehicle leaves the network, its socket will be closed and the `in_use` parameter set to "False". In order to use the same vehicle node structure object for a different vehicle, we bind the socket to an address and re-enable 'Recv' calls on it.

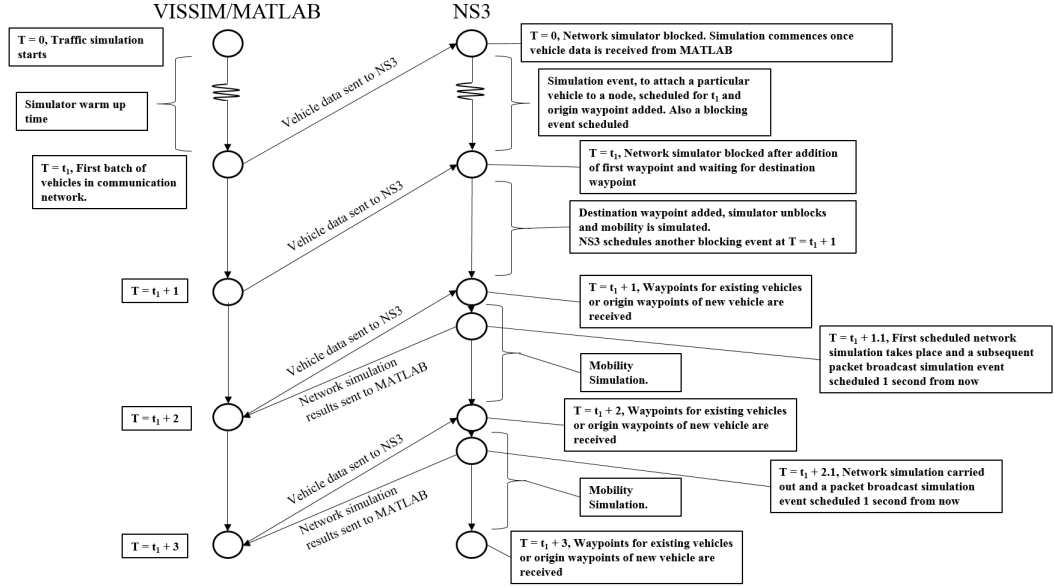


Figure 2: Flow of data amongst simulators

3.3 Selection of the appropriate mobility model in NS3

In order to model the mobility of the vehicles accurately, we chose the Waypoint Mobility Model in NS3. To provide waypoints in an online manner, we decided to run VISSIM, one simulation step in advance, as has been described in [9]. This ensured that both origin and destination waypoints are always available to simulate node mobility.

4 Simulation setup

To test the simulator, we decided to model the GLOSA application, concerned with communicating Traffic Signal information (current phase, the phase remaining time and position of the traffic signal), to oncoming vehicles by a road-side-unit (RSU). The code for GLOSA was written in MATLAB, which used the simulation results from NS3. NS3 was used to carry out network simulations between the RSU and the vehicles and provided results on whether a packet broadcast by the RSU was received by the oncoming vehicles or dropped. Based on the outcome, the velocity change instructions were given to the appropriate vehicles in VISSIM.

4.1 Traffic Modelling

The traffic network in VISSIM is based on the three intersection Alexandra Road (Fig 3a) corridor in Singapore. For the purpose of V2X simulations, only one traffic signal head (with fixed-time signal control) on one of intersections is assumed to be equipped with an RSU. Also, it is assumed that all vehicles that are on the link served by the RSU, are equipped with On-Board Units (OBU) and can identify the relevant signal head. The middle intersection was chosen and the application was tested on all vehicles travelling from West to East, through the

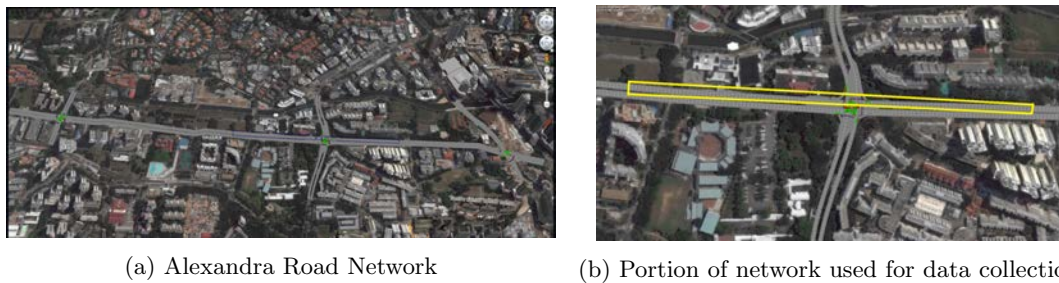


Figure 3: Traffic Network

intersection. To have a high vehicle density on this route, vehicle inputs of 6000 vph, 1000 vph and 1000 vph were given to the three network inlets in the west.

A section of the above-mentioned link along with a section of the subsequent link that it is connected to was chosen as the area for data collection, Fig 3b, which includes the afore-mentioned traffic intersection. The length of this section is approximately 650 m. Two simulations were carried out, with GLOSA and without GLOSA. The random seed was kept the same for both the simulations in order to ensure the vehicle flow pattern is as similar as possible. The minimum speed limit, that GLOSA can assign, was chosen to be 25 km/hr, while the maximum to be 85 km/hr. The parameters being monitored are total number of vehicles that pass through the section, average vehicle delay, average stopped delay, average number of vehicle stops, total Carbon Monoxide (CO) emissions and total fuel consumption. The simulations are run for a total of 600 simulation seconds.

4.2 Communication simulation

In order to simulate communication between the RSU and vehicles, it was decided to broadcast a packet of 1000 bytes every 1 second. Also, to keep in accordance with the channel assignment policies of DSRC WAVE, we decided to utilize the high power traffic efficiency service channel 184, for which the Effective Isotropic Radiated Power (EIRP) is 40 dBm [11],[10]. Unlike [27], implementation of the the network simulator NS3 helps one to get the correct metrics on what is the actual network coverage based on the chosen transmit power and using the most accurate model of the DSRC WAVE architecture. Therefore, using the integrated platform one can check the performance of GLOSA or any other non-safety application with various transmit power levels and use it for determining the optimal EIRP value. In our GLOSA application, simulations were carried out to determine which vehicles are within the coverage range, at every broadcast interval, based on their mobility information. Using a transmit power of 40 dBm, it was found GLOSA advisories could be communicated to vehicles at a maximum distance of 580 meters (assuming clear line-of-sight). Therefore, a message broadcast rate of 1 Hz was deemed more than sufficient to indicate vehicles about the desired speed. Also, providing advisories at greater distances from the RSU ensures vehicle have enough time to accelerate/decelerate to the desired speed.

Once network simulations are complete, the results conveyed to MATLAB are the numbers of those vehicles which received the packet and the delay in reception.

Vehicle waypoints are supplied for the T -th and $(T+1)$ -th second while a broadcast simulation is scheduled for every $(T+0.1)$ seconds. The network simulation results will be com-

municated back to MATLAB, which is currently blocked, at the $(T+1)$ -th second. In order to accommodate for this lag, we subtract $T+0.1$ (instead of $T+1$) from the current signal phase time (to calculate remaining phase time), assuming that is the signal timing information conveyed by the traffic signal. The speed advisory is implemented at $T+1$, which incorporates a small delay in advisory assignment. However, a human driver may take about a second to react to a visual stimuli, like a GLOSA announcement on a HMI screen. Therefore, the delay in assignment is considered justified in the present scope of the simulations.

5 Results

Table 1 shows the results for the two kinds of simulations carried out. From the results, it can be observed that implementation of GLOSA certainly offers an improvement with respect to traffic efficiency. For instance, average stopped delay, which is the time a vehicle is stopped in queue while waiting to pass through the intersection, reduces and also the average vehicle stops per vehicle by at least 50% each. There is a total decrease in fuel consumption by approximately 7.4% and a reduction of approximately 7% in average Carbon Monoxide emissions. Also, it can be gathered from the simulations there is a higher amount of network throughput (122 vehicles) when using GLOSA as against 99 vehicles without using GLOSA, for the same segment of 650 m that is being monitored and for a simulation time of 600 seconds.

	Total vehicles observed	Average vehicle delay (Seconds)	Average stopped delay (Seconds)	Average vehicle stops (Per vehicle)	Carbon Monoxide emissions (Grams)	Fuel consumption (Litres/100KM)
Without GLOSA	99	15.29	8.65	0.51	1.73	14.5
With GLOSA	122	15.08	3.69	0.25	1.608	13.4

Table 1: Simulation results

6 Conclusion and Future Work

The results shown above have been compared with the relevant results in [16], which reports a 7% reduction in fuel consumption and with [27], which reports a reduction of 50% in vehicle stopped delay, and has been found to be giving similar results.

The objective of this endeavour was to demonstrate that the three specific simulators can be coupled in a holistic manner from which comprehensive data can be derived and which can help in the test, design and validation of V2X applications. An alternative technique of coupling the three simulators would be to use principles of High Level Architecture (HLA) and a Runtime Infrastructure (RTI) for data exchange and simulator synchronization. Our approach is loosely inspired by the HLA structure, where MATLAB acts as the de facto RTI and is responsible for duties such as time synchronization along with object and data management. However, to improve flexibility of the simulation environment, so that one or more of the simulators can be changed to a different tool, or if it is desired that an additional simulator be added to the framework, we plan on using a generic RTI. Since the platform will be used for carrying out

large scale simulations in the future, using an RTI to split the traffic or network simulations into smaller chunks will make the simulation tool computationally efficient and easily scalable. To this effect, we plan on writing wrapper libraries for VISSIM and MATLAB that allow RTI ambassador methods to be called and also implement the federate ambassador, for the purpose of communicating data to the simulators. This will make the RTI implementation responsible for data exchange and MATLAB can simply be used for running the application algorithms.

It has been envisioned to expand this simulator further in order to carry out simulation of multiple applications. One of our next steps, in this regards, would be to model platooning of vehicles that are connected via a V2V communication network. We aim to analyse how a platoon can improve movement of vehicles, especially commercial vehicles for logistic operations and whether it can offer significant benefits from a business point of view. We will be exploring the necessary Cooperative Adaptive Cruise Control strategies, to realize platooning, and simulating them on this platform to evaluate their performance. We will also explore alternatives that will allow us to do away with the 1 second advisory assignment delay.

References

- [1] EDB singapore. <https://www.edb.gov.sg/content/edb/en.html>.
- [2] The Mathworks. <http://www.mathworks.com/>.
- [3] Network Simulator-3. <https://www.nsnam.org/>.
- [4] NTU-NXP smart mobility test bed. <http://media.ntu.edu.sg/NewsReleases/Pages/newsdetail.aspx?news=e91a6458-93ce-4b48-a891-9a64e76f38fa>.
- [5] NXP Semiconductors. http://www.nxp.com/about/about-nxp:COMPANY_INFO_HOME.
- [6] PTV VISSIM. <http://vision-traffic.ptvgroup.com/en-us/products/ptv-vissim/>.
- [7] Raphael Bialon. Coupled simulation of road traffic, v2x-applications and v2x-communication via mobile cellular networks. 2015.
- [8] David R Choffnes and Fabián 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*, pages 69–78. ACM, 2005.
- [9] Stephan Eichler, Benedikt Ostermaier, Christoph Schroth, and Timo Kosch. Simulation of car-to-car messaging: Analyzing the impact on road traffic. In *null*, pages 507–510. IEEE, 2005.
- [10] Stephan U Eichler. Performance evaluation of the ieee 802.11 p wave communication standard. In *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th*, pages 2199–2203. IEEE, 2007.
- [11] Sebastian Gräfling, Petri Mahonen, and Janne Riihijarvi. Performance evaluation of ieee 1609 wave and ieee 802.11 p for vehicular communications. In *Ubiquitous and Future Networks (ICUFN), 2010 Second International Conference on*, pages 344–348. IEEE, 2010.
- [12] MR Hafner, D Cunningham, L Caminiti, and D Del Vecchio. Automated vehicle-to-vehicle collision avoidance at intersections. In *Proceedings of World Congress on Intelligent Transport Systems*, 2011.
- [13] Jérôme Härri, Fethi Filali, Christian Bonnet, and Marco Fiore. Vanetmobisim: generating realistic mobility patterns for vanets. In *Proceedings of the 3rd international workshop on Vehicular ad hoc networks*, pages 96–97. ACM, 2006.
- [14] Daniel Jiang and Luca Delgrossi. Ieee 802.11 p: Towards an international standard for wireless access in vehicular environments. In *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*, pages 2036–2040. IEEE, 2008.

- [15] Feliz Kristianto Karnadi, Zhi Hai Mo, and Kun-chan Lan. Rapid generation of realistic mobility models for vanet. In *Wireless Communications and Networking Conference, 2007. WCNC 2007. IEEE*, pages 2506–2511. IEEE, 2007.
- [16] Konstantinos Katsaros, Ralf Kernchen, Mehrdad Dianati, and David Rieck. Performance study of a green light optimized speed advisory (glosa) application using an integrated cooperative its simulation platform. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2011 7th International*, pages 918–923. IEEE, 2011.
- [17] Johan Kjellberg. *Implementing control algorithms for platooning based on V2V communication*. PhD thesis, master thesis, Royal Institute of Technology, 2011.
- [18] Daniel Krajzewicz, Laura Bieker, Jérôme Härri, and Robbin Blokpoel. Simulation of v2x applications with the itetris system. *Procedia-Social and Behavioral Sciences*, 48:1482–1492, 2012.
- [19] Christian Lochert, Andreas Barthels, Alfonso Cervantes, Martin Mauve, and Murat Caliskan. Multiple simulator interlinking environment for ivc. In *Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks*, pages 87–88. ACM, 2005.
- [20] Rahul Mangharam, Daniel S Weller, Daniel D Stancil, Ragunathan Rajkumar, and Jayendra 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*, pages 59–68. ACM, 2005.
- [21] Hamed Noori. Realistic urban traffic simulation as vehicular ad-hoc network (vanet) via veins framework. In *12th Conference of Open Innovations Framework Programm. FRUCT*, 2012.
- [22] Hyungjun Park, Adelin Miloslavov, Joyoung Lee, Malathi Veeraraghavan, Byungkyu Park, and Brian Smith. Integrated traffic-communication simulation evaluation environment for intelligdrive applications using sae j2735 message sets. *Transportation Research Record: Journal of the Transportation Research Board*, (2243):117–126, 2011.
- [23] Michal Piorkowski, Maxim Raya, A Lezama Lugo, Panagiotis Papadimitratos, Matthias Grossglauser, and J-P Hubaux. Trans: realistic joint traffic and network simulator for vanets. *ACM SIGMOBILE Mobile Computing and Communications Review*, 12(1):31–33, 2008.
- [24] Robert Protzmann, Kim Mahler, Konstantin Oltmann, and Ilja Radusch. Extending the v2x simulation environment vsimrti with advanced communication models. In *ITS Telecommunications (ITST), 2012 12th International Conference on*, pages 683–688. IEEE, 2012.
- [25] Tobias Queck, Bjorn Schunemann, Ilja Radusch, and Christoph Meinel. Realistic simulation of v2x communication scenarios. In *Asia-Pacific Services Computing Conference, 2008. APSCC’08. IEEE*, pages 1623–1627. IEEE, 2008.
- [26] Kaveh Shafiee, Jinwoo Brian Lee, Victor Leung, and Garland Chow. Modeling and simulation of vehicular networks. In *Proceedings of the first ACM international symposium on Design and analysis of intelligent vehicular networks and applications*, pages 77–86. ACM, 2011.
- [27] Aleksandar Stevanovic, Jelka Stevanovic, and Cameron Kergaye. Impact of signal phasing information accuracy on green light optimized speed advisory systems. In *Proc. 92nd Annual Meeting of the Transportation Research Board (TRB), Washington DC, USA*, 2013.
- [28] R Uzcategui and Guillermo Acosta-Marum. Wave: a tutorial. *Communications Magazine, IEEE*, 47(5):126–133, 2009.
- [29] SY Wang, CL Chou, YH Chiu, YS Tzeng, MS Hsu, YW Cheng, WL Liu, and TW Ho. Nctuns 4.0: an integrated simulation platform for vehicular traffic, communication, and network researches. In *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th*, pages 2081–2085. IEEE, 2007.
- [30] Jan W Wedel, Bjorn Schunemann, and Ilja Radusch. V2x-based traffic congestion recognition and avoidance. In *Pervasive Systems, Algorithms, and Networks (ISPAN), 2009 10th International Symposium on*, pages 637–641. IEEE, 2009.
- [31] Christian Weiß. V2x communication in europe—from research projects towards standardization and field testing of vehicle communication technology. *Computer Networks*, 55(14):3103–3119, 2011.

- [32] Fei Ye, Matthew Adams, and Sumit Roy. V2v wireless communication protocol for rear-end collision avoidance on highways. In *Communications Workshops, 2008. ICC Workshops' 08. IEEE International Conference on*, pages 375–379. IEEE, 2008.