A High-Level Architecture SimIVC for Simulating the Traffic Network

Xiaowei XU[1], Tao JIANG[1], Pengfei LI[2], Tony QIU[2], Jingyuan WANG[1]
Xuecheng ZOU[1], Yu HU[3]

[1] School of Optical and Electronic Information, Huazhong University of Science and Technology, Wuhan, Hubei province, 430074

[2] Department of Civil and Environmental Engineering, University of Alberta, Alberta, T6G2W2, Canada

[3] Corresponding author, email: bryanhu@hust.edu.cn, Wuhan Research Institute for Smarter Cities, Wuhan, Hubei province, 430074, email: bryanhu@hust.edu.cn

ABSTRACT

Recently, much effort has been devoted to advancing Connected Vehicle technology. It is foreseeable that vehicles operating in networks will come to characterize our everyday life in the near future. Nevertheless, it is not yet practical to conduct large-scale experiments in the field for researching Inter-Vehicle Communication (IVC) protocols and their effects on traffic. For this reason, IVC simulation is in high demand as a means for evaluating the performance and applicability of existing network protocols and their impact on traffic. To simulate both vehicular driving behavior and networking protocols with precision a state-of-the-art simulator must be developed that is capable of both fine-grained traffic microscopic simulation and network simulation. We therefore developed a High-Level Architecture (HLA) simulator called SimIVC (Simulator for Inter-Vehicle Communication), composed of a traffic simulator, PTV VISSIM, and a network simulator, OMNeT++. A study was conducted to show the effectiveness of SimIVC. In the study, two network protocols (DSRC and Wi-Fi) were evaluated in a well-calibrated road simulation network developed using the HLA SimIVC.

1. INTRODUCTION

Realistic simulation of Inter-Vehicle Communication (IVC) protocols is one of the main challenges facing the Cooperative Vehicle-Infrastructure System (CVIS) research domain. Much effort has been devoted to advancing research into and development of CVIS simulators. For example, Veins (Sommer et al., 2011) is an IVC simulation framework composed of an event-based network simulator, OMNeT++ (Varga, 2001) , and a road traffic simulation package, SUMO (Behrisch et al., 2011) . Some effort has also been made to integrate the traffic simulator, VISSIM, with the network simulator, ns-2 (Sommer et al., 2008). And the integration of VISSIM and NUTUns is done in (Miloslavov et al., 2010). However, VISSIM and OMNeT++, the top-ranking tools from both domains(Weingartner et al., 2009; Ratrout and Rahman, 2009), have not been integrated into a single simulator before. The HLA SimIVC fills this gap.

HLA SimIVC accelerates and facilitates the evaluation of network protocols within an inter-vehicular environment in the following ways. First, SimIVC provides an effective integration of the traffic simulator VISSIM and network simulator OMNeT++. This kind of integration, which is also called bidirectional coupling (Köpke et al., 2008) , has two main benefits: 1) it provides greater insight into both the traffic and network simulation processes; 2) because VISSIM and OMNeT++ are both top-ranking tools from both domains, the bidirectional coupling simulation result can be more realistic (Sommer et al., 2008).

Second, as to the network simulator element, SimIVC integrates two verified OMNeT++ extensions for the wireless network protocols: Dedicated Short-Range Communication (DSRC) from Veins (Veins 2.0, 2012) and Wi-Fi 802.11g from INET (INET 2.0.0, 2012).

In the following sections, the general structure and implementation details of SimIVC are presented and the effectiveness of SimIVC is demonstrated through an example of the delay performance of both the DSRC and Wi-Fi 802.11g wireless protocols within IVC environment.

2. SIMULATION PLATFORM

The general structure and implementation details of SimIVC are presented in this section.

2.1 Platform General Structure

The general structure of the HLA is illustrated in Figure 1.

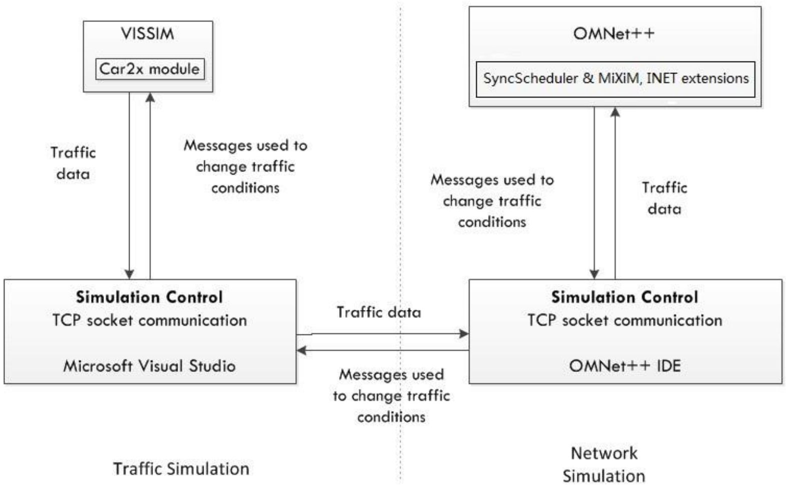

Figure 1: General structure of SimIVC

As shown in Figure 1, a Simulation Control Layer (SimCL) is created to a) handle synchronization between the traffic simulator, VISSIM, and the network simulator, OMNeT++, and b) transfer traffic information from VISSIM to OMNeT++ and transfer information back again for traffic guidance and management after calculations are carried out by OMNeT++. The implementation details are described in the next section.

2.2 Platform Implementation

2.2.1 Traffic Simulator

The essential task of SimCL is to interact with VISSIM, which means VISSIM must expose some controllable interfaces to accomplish the interaction. The requirements for interaction interfaces are: 1) that they can not only obtain information from VISSIM, but can also control the vehicle's behavior within the VISSIM road network; 2) the more light-weight, the better; 3) that they cannot conflict with the internal Wiedemann driving behavior model (Wiedemann, 1974) of VISSIM -- otherwise, the user will have to implement a separate driving behavior model.

With these requirements in mind, two approaches are evaluated. One concerns the *car2x module,* which is officially included with VISSIM; the other concerns the *external driver model* functionality of VISSIM.

Though the external driver model is more flexible and more light-weight than the car2x module, the lack of Wiedemann driving behavior makes it hard to utilize the Driving Behavior functionality of VISSIM. Though the car2x module is not as light-weight as its counterpart, it is more concise. Its logic is entirely implemented within a function named *processTimestep,* which can be written in either Python or C++, and is able to simulate every time step in order to process the data and perform operations on vehicles more precisely.

The deficiency of the car2x module is that the wireless communication component among vehicles is implemented using a proprietary component, VCOM, which cannot be modified despite its low simulation precision.

After comparing the two approaches, the *former* one (car2x module) is chosen. Though its reliance upon VCOM is not desirable, it can simply be bypassed with the help of OMNeT++, a highly-customizable network simulator, which will be described in the next section.

2.2.2 Network Simulator

OMNeT++ is a discrete-events system simulator. The model topology is described with NED language in *ned* files. In ned files, the final module to be simulated is called a *network*. A *network* consists of *simple* modules and *compound* modules, the latter of which consists of *simple* modules and channels, etc. Simple modules are active modules written in C++, which handle essential tasks such as performing a specific task after receiving a message in a specific form.

In SimIVC, each vehicle is represented as a *host* compound module within

OMNeT++. The *host* is partitioned into several sub-modules; the principle behind the module partition follows the classic and widely used TCP/IP protocol suite (i.e., Physical Layer (PHY) module, Medium Access Control (MAC) layer module, network layer module, application layer module), but with several modifications according to each wireless protocol, which will be described below.

The typical protocol suite for DSRC are as follows: IEEE 802.11p (based on Wi-Fi 802.11a) for PHY, IEEE 1609.3 & 1609.4 for MAC, and IEEE 802.2 for Logic Link Control (LLC). WAVE Short Message Protocol (WSMP), defined by ASTM International, is used to replace traditional IP and UDP/TCP. The DSRC implementation integrated in SimIVC derives from MiXiM project (MiXiM 2.2.1, 2012).

Wi-Fi 802.11g is an amendment of the IEEE 802.11 specification and uses the same Orthogonal Frequency-Division Multiplexing (OFDM) modulation scheme as 802.11a. The typical 802.11g protocol implementation is much more like traditional TCP/IP, in which the PHY is defined in 802.11g and the MAC and authentication process are defined by the IEEE 802.11 specification (IEEE Standard, 1999). The 802.11g implementation in SimIVC derives from INET project .

2.2.3 Simulation Control Layer

Due to the nature of bi-directional coupling in HLA simulation, synchronization between the traffic simulator and the network simulator is a prerequisite functionality. The VISSIM and OMNeT++ frameworks are both discrete-events simulators, each with its own internal simulation time step. This feature greatly eases the development of a synchronization component for SimIVC because the synchronization between them is simply a synchronization between each internal simulation time step as shown in Figure 2.

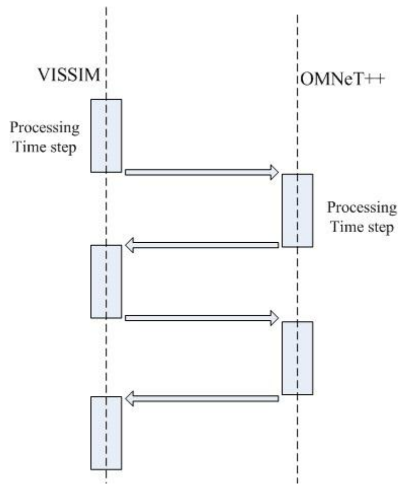

Figure 2: Sequence diagram of message exchanging and synchronization

To facilitate the interaction between traffic and network simulators, a socket communication interface is implemented with Boost Serialization (Kambadur et al.,

2006) and the Asio library (asio-1.5.3, 2012).

Through this interface, the OMNeT++ network simulator can receive information about a) the vehicles that are currently active within the VISSIM road network, b) the position and speed of each active vehicle, and c) which vehicles have entered or exited the network. With this information, OMNeT++ can learn about the communication status of vehicles through calculation.

As the counterpart to OMNeT++, the VISSIM traffic simulator sends information to OMNeT++, as described above, and receives feedback messages (i.e., the communication status among the connected vehicles) for future traffic guidance and management.

3 APPLICATION MODEL AND SIMULATION

In the following, the effectiveness of SimIVC is demonstrated through an example reflecting the delay performance of the DSRC and Wi-Fi 802.11g wireless protocols within the IVC environment.

3.1 General Model Description

The road network adopted in the example is a well-calibrated network based on Whitemud Drive, the main east-west freeway in southern Edmonton, Alberta, Canada. As shown in Figure 3, the road network has been carefully modeled and calibrated with VISSIM (Tony, 2012).

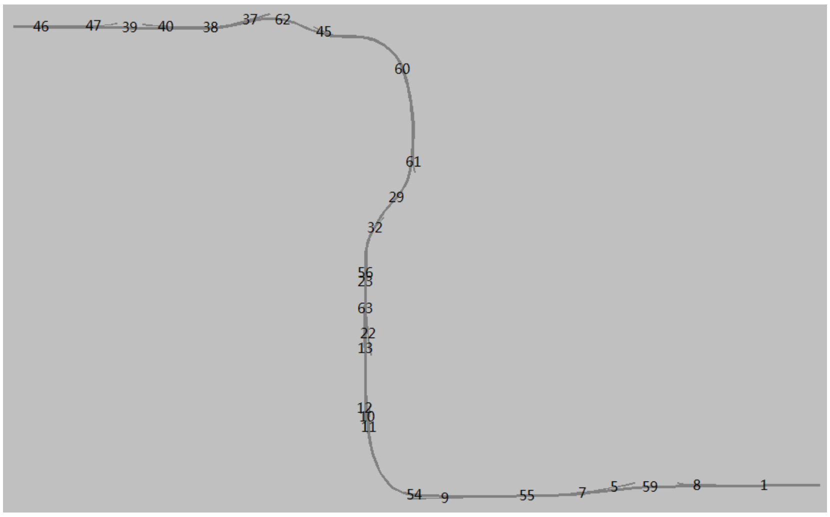

Figure 3: Whitemud Drive, modeled with VISSIM

3.2 Simulation Setup

In the experiment, an incident occurred within road sector 60 as shown in Figure 3, which is modeled as a parking lot with a capacity of only one. After the incident occurs, a traffic jam develops, and the incident vehicle begins to broadcast a notification message to other vehicles within the *maximum communication range* (this range is calculated by OMNeT++). It is noteworthy that not all the vehicles within the maximum communication range can receive the broadcast message because a signal can only be recognized as an effective message -- and not as noise -- when the received Signal to Noise Ratio (SNR) goes beyond the SNR reception threshold, which is a host-specific parameter.

Two different, but typical, topologies are adopted with regards to DSRC and Wi-Fi. In the DSRC experiment, all the vehicles are communicating directly with each other, i.e., in Vehicle-to-Vehicle (V2V) mode. For simulating data services, vehicles within the DSRC network begin broadcasting messages periodically at stochastic intervals after entering the network, with random packet sizes ranging from 8 bytes to 1395 bytes (Yin et al., 2004). After the incident occurs, the incident vehicle starts to broadcast the same 100-byte notification message periodically at stochastic intervals ranging between [0.95, 1.05], i.e. within the upper and lower bounds of 1±0.05 seconds. The random interval shifting within the 0.05s (50ms) boundary corresponds to the channel switching mechanism of the DSRC (*ASTM* , *2003*).

For Wi-Fi, vehicles communicate via a wireless Access Point (AP). In this experiment, there is only one AP, which is located beside the incident vehicle. The vehicles are not allowed to communicate directly with each other, but can only connect to the AP, through which messages are forwarded. Before a vehicle can send or receive a message, it should interact with the AP in order to go through the Wi-Fi authentication process successfully. All of the authenticated vehicles, as is also the case with the DSRC experiment, communicate periodically at stochastic intervals with the AP after being authenticated and, again, with random packet sizes ranging from 8B to 1395B. Likewise, the incident vehicle starts to broadcast the same 100-byte notification message periodically at the same stochastic interval as was used in the DSRC experiment above. And in Wi-Fi authentication process, each node has to scan 3 channels continuously until it gets the right channel the AP works and starts to send authentication request.

More detailed parameters are shown in Table.1 below. All the other parameters are default in the MiXiM project and the INet project, in which it is to be noted that the physical layer model was highly abstract and did not account for multi-path delay spread and Doppler effect [18].

Table.1 Detailed parameters of protocols

	DSRC	WiFi(802.11g)
Transmission power	100mW, 20mW	100mW, 20mW
Receiving sensitivity	-82dBm	-72dBm
Thermal noise	-110dBm	-110dBm
Data rate	18Mbps	24Mbps

4. RESULTS AND DISCUSSION

For DSRC and Wi-Fi 802.11g, the principles behind the calculation of delay of a single packet are the same, as shown in (4.1). $T_{incident}$ and $T_{receive}$ are time stamps for incident happening and packet receiving, respectively.

$$Delay = T_{receive} - T_{incident} \tag{4.1}$$

Depending on the transmission power of the incident vehicle (IV), formula (4.2) is used to calculate the final result. In the formula, *i* denotes the total packet numbers sent by incident vehicle within a 100 seconds duration, *j* denotes the number of vehicles that received the notification message for each packet, and $delay_{ji}$ denotes any delay for the l_{th} vehicle regarding the k_{th} packet.

$$averageDelay = \left(\sum_{k=1}^{i} \left(\frac{\sum_{l=1}^{j} delay_{j_i}}{j} \right) \right) / i \tag{4.2}$$

In the case of Wi-Fi 802.11g, the final results are divided into two parts, T_{in} and T_{out}, due to the authentication process. T_{in} denotes the average delay of packets received by vehicles that have been authenticated prior to packet reception (the calculation method is exactly the same with the DSRC experiment as above, using the same time duration of 100 seconds). T_{out} denotes the average delay of packets received by vehicles that are being authenticated prior to their *first* packet reception, i.e., the $delay_{ij}$ should be calculated as in (4.3), where $T_{startingAuth}$ denotes the time stamp of a vehicle starting to interact with AP for authenticating and $T_{firstPacketReception}$ denotes the time stamp for its first packet reception from AP.

$$delay_{i_j (T_{out})} = T_{firstPacketReception} - T_{startingAuth} \tag{4.3}$$

The WiFi protocol's delays under different degrees of power are shown in Figure 4. When the network load is low, T_{in} and T_{out} are both low and do not increase

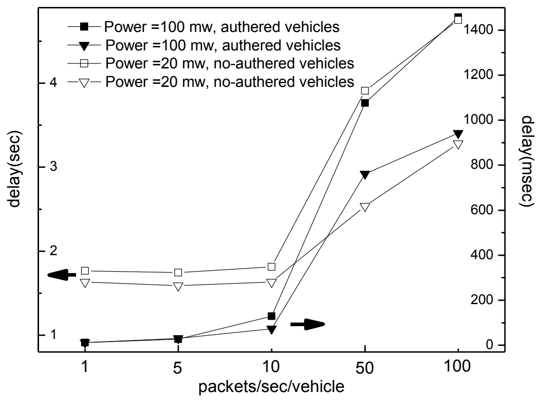

Figure.4 WiFi's delays verses different powers

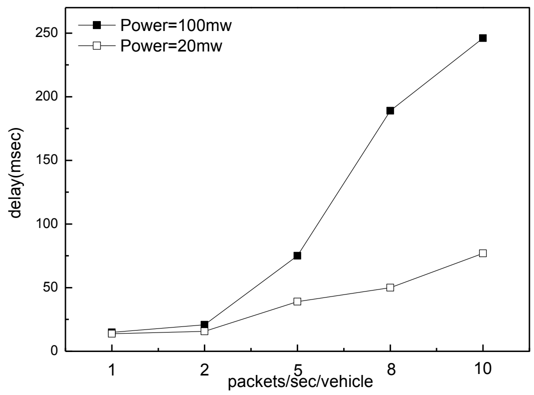

Figure.5 DSRC's delays verses different powers

much as the network load gets higher. As the network load gets higher, both T_{in} and T_{out} increase quickly to approximately 1 second. Due to the authentication process, the T_{out} is always greater than 1 second. Thus it can be easily deduced that the delay with a larger power will be greater than the one with a smaller power.

Figure 5 shows the DSRC's delays under different degrees of power. It can be seen that, as the network load increases, the delay increases. The delay also increases more quickly with a heavier network load.

From the two figures, it is evident that the WiFi's T_{in} delays can match the DSRC's delays when the network load is low; however, for WiFi, the majority of vehicles are subjected to the T_{out} delay for authenticating vehicles. The T_{out} delay for WiFi is always greater than 1 second, which has no comparability with the DSRC's delay. DSRC is a better solution for the application because of its low delay. If the authentication process can be avoided in WiFi, then the delay may decrease immensely.

5.CONCLUSION

In this paper, we introduce HLA SimIVC, an effective tool for facilitating the evaluation of wireless network protocols within inter-vehicular environments, and demonstrate its effectiveness though an example of the delay performances of DSRC and Wi-Fi 802.11g protocols. Though still in development, SimIVC is already a powerful tool for helping researchers utilize the merits of both VISSIM and OMNeT++ to perform research on IVC network protocols.

REFERENCES

Sommer C., German R. and Dressler F. (2011). "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis", *IEEE Transactions on Mobile Computing*, vol. 10(1), January 2011. 3-15.

Varga A. (2001). "The OMNeT++ Discrete Event Simulation System", *European Simulation Multiconference* (ESM2001). June 2001.

Behrisch M., Bieker L., Erdmann J., and Krajzewicz D. (2011) . "SUMO-Simulation Of Urban MObility: An Overview", SIMUL 2011, *The Third International Conference on Advances in System Simulation,* 2011, 63-68.

Sommer C., Dietrich I. And Dressler F. (2008). "A Simulation Model of DYMO for Ad Hoc Routing in OMNeT++", *1st ACM/ICST International Conference on Simulation Tools and Techniques for Communications, Networks and Systems (SIMUTools 2008)*: 1*st ACM/ICST International Workshop on OMNeT++ (OMNeT++ 2008)*. Marseille, France: ACM, March 2008.

Miloslavov A., Park H., Lee J., Veeraraghavan M., Park B. and Brian L. (2010). "Details of the integrated VISSIM-NCTUns vehicular-wireless simulation platform", Technical report, University of Virginia, 2010.

Weingartner E., vom Lehn H., and Wehrle K. (2009). “A performance comparison of recent network simulators”, *In Proceedings of the IEEE International Conference on Communications 2009* , Dresden, Germany, IEEE, June 2009.

Ratrout, N.T., Rahman, S.M. (2009). “A comparative analysis of currently used microscopic and macroscopic traffic simulation software”, *The Arabian Journal for Science and Engineering*, 2009, vol. 34, no. 1B, 121-133.

Köpke A., Swigulski M., Wessel K., Willkomm D., et al.(2008). “Simulating wireless and mobile networks in OMNeT++ the MiXiM vision”,*Simutools'08*, article No. 71. 2008.

Sommer C., Yao Z., German R.,and Dressler F.(2008).“On the Need for Bidirectional Coupling of Road Traffic Microsimulation and Network Simulation,” *Proc. ACM MobiHoc: First Int'l Workshop Mobility Models for Networking Research (MobilityModels '08)*, 2008, 41-48.

Veins 2.0, http://veins.car2x.org/download/, (Dec.22,2012).

INET 2.0.0, http://inet.omnetpp.org/, (Dec.22,2012).

Wiedemann R.(1974). “Simulation des Strassenverkehrsusses (Simulation of road traffic flow, in German)”, *Schriftenreihe des Instituts fÄur Verkehrswesen*, University Karlsruhe, Karl-sruhe, Germany, 1974.

MiXiM 2.2.1, http://sourceforge.net/projects/mixim/files/mixim/,(Dec.20, 2012).

IEEE Standard, (1999).“Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band”

Kambadur P., Gregor D., Lumsdaine A., and Dharurkar A. (2006). “Modernizing the C++ interface to MPI”, *13th European PVM/MPI Users Group Meeting (Euro PVM/MPI'06)*. Springer, 2006.

asio-1.5.3**,** http://sourceforge.net/projects/asio/files/**,** **(**Dec.20, 2012**)**.

Tony Z. Qiu, “Online simulation of networked vehicle based active traffic management for freeway operation”, Final Report, University of Alberta, 2012.

Yin J., ElBatt T., Yeung G., Ryu B., Habermas S., Krishnan H., and Talty T. (2004). “Performance evaluation of safety applications over DSRC vehicular ad hoc networks”. *Proc. of ACM workshop on Vehicular Ad Hoc Networks (VANET),* 2004.

ASTM E2213-03, (2003). “Standard Specification for Telecommunications and Information Exchange Between Roadside and Vehicle Systems-5GHz Band Dedicated Short Range Communications (DSRC) Medium Access Control (MAC) and Physical Layer (PHY) Specifications”