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 reconstructed, 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 has garnered a considerable amount of attention in the past decade for the purpose of improving traffic safety and efficiency. The creation of
a the IEEE 802.11p standard and dedication of the 5.9 GHz frequency bandwidth specifically for supporting communications for ITS applications has motivated many countries to explore 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 which will offer researchers the ideal playground to test V2X technologies and applications. This project is supported by the Economic Development Board of Singapore and is being carried out by Nanyang Technological University and NXP semi-conductors in a joint collaboration. One part of this test-bed project is to develop a simulator that is capable of modelling and testing V2X communication and applications in areas beyond the test bed and 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, each with a particular capability, that when operated together may allow one to analyse the effects of implementing V2X technologies in actual traffic situations. Following are the three major components that need to be accurately represented by any integrated simulation environment:

- **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 VISSIM, for traffic simulation, NS3 for network simulation and MATLAB 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) 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. This platform will allow testing of the algorithms on a realistic simulation platform, without having to convert it to a different programming language. Also, for later implementation on the test bed, C/C++ code can be generated using the Simulink Coder.
2 Literature Survey

Specifically, there are two ways to go about simulating V2X operations as elucidated in [12]. These are classified as a network-centric approach and an application-centric approach. The network centric approach pertains to offline analysis and helps in testing V2X protocols and does not allow real-time modification of vehicle/traffic behaviour based on exchange of messages.

The application-centric approach on the other hand pertains to a much broader set of use-cases (safety application, vehicle-rerouting), i.e., applications whose information will affect the future movement of the vehicles. This approach is concerned with online analysis where real-time communication between simulators is required.

Specifically, a review of those works which emphasize on an application-centric based simulation environment was carried out before embarking on this project. For references on network-centric approaches please look into [2], [4], [9], [5].

One of the most widely used simulator coupling environments available is VsimRTI [14], [13] 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 endeavors for integration of simulators for modelling V2X operations. Some of the simulators that have been coupled using VsimRTI are SUMO and Jist/SWans [6], MATLAB CCMSim (Car2X Channel Model SIMulator) and OMNET++ [13] and SUMO and OMNET++ [1], [13]. 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 [8] 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 Integrated Wireless and Traffic Platform for RealTime Road Traffic Management Solutions (iTetris) simulator [7], which was an European Union Framework Program 7 funded project. It aims at combining SUMO and NS3 and the novelty lies with the design of a central control component called the iTETRIS Control System (iCS) that enables coordination between the two simulators. In addition, there exists the simulator integration environment called Traffic and Network Simulation Environment (TraNS) [12] which links two open source simulators: SUMO and NS2. TraNS has the ability to operate in both offline and online mode. However, development of TraNS has ceased and does not support newer versions of SUMO or NS3. Both iTetris and traNS are not suitable for our purposes since they use SUMO for traffic modelling. Apart from these environments there exists the Veins framework which is an integration platform that couples SUMO and the INET framework from OMNeT++ [10] and [3] whose authors have utilized their custom traffic simulator called CARISMA and combined it with NS2. Furthermore, there also exists the NCTUins 4.0 environment [17]. 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 realized and tested, as has also been mentioned in both [12] and [15]. However this platform was extended by coupling with VISSIM [11], which also inculeated the applications-dedicated message sets, SAE J2735.
3 Simulator Design

Fig 1 shows the basic simulator coupling idea. Since NS3 has been developed for Linux while VISSIM can be used only on Windows, we used a virtual machine with Linux, installed NS3 on it and linked the host machine(with Windows) to the virtual machine via a virtual network. We then used sockets for communication of data between VISSIM and NS3(MATLAB is also installed on Windows).

![Block diagram of the simulation environment](image)

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 follow

- **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 being run in a loop using the MATLAB COM interface and every iteration updates the simulation by 0.1 second. On the other hand, NS3 is a discrete-event simulator which means that simulator time update is non-linear.

- **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, 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 of NS3 which is the most appropriate for representing VANETs.

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. As soon as 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 vehicle enter the communication network, and their position information is sent to NS3(queryed 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 the VISSIM simulation, which is currently blocked (through MATLAB) and is awaiting simulation results from NS3. Once communication simulation commences, a blocking event is scheduled every simulation second to wait for data and the traffic simulation code in MATLAB ensures that every second, vehicle data from the communication network area is sent to NS3. A flowchart to illustrate 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 to them. Sockets will be attached to only those many number of nodes as there are vehicles in the communication network. In order to characterize a vehicle in NS3, we created a Vehicle Node structure which contains the following information:

```c
struct veh_Nodes
{
    Ptr<Node> veh_node; // A smart pointer to the node object
    Ptr<Socket> veh_socket; // A smart pointer to the socket that is 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 or not
};
```

A vector of such vehicle structures objects are created (same as the number of nodes) and each node 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 particular Vehicle Node structure object, which is currently not ‘in use’ and a socket will attached. This step is carried out for every new vehicle that enters the network in subsequent simulation steps. Once a Vehicle Node is active, only its mobility model is updated every simulation second, by comparing the the `veh_number` member of the object with the data received from MATLAB. To discontinue communication once the vehicle leaves the network, its socket will be closed and the `in_use` parameter set to 0. In order to use the same node structure (with an existing closed socket) for a different vehicle, we bind the socket to an address and re-enable ‘Recv’ calls on the socket again. This allows re-cycling the same nodes for different vehicles.

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 the traffic simulations, one simulation step in advance, as has also been described in [3]. This ensured that a destination waypoint is always available, in NS3, for the node to simulate mobility between two consecutive points.

4 Simulation setup

To perform validation of above design, we decided to simulate the Green Light Optimized Speed Advisory (GLOSA application, which is concerned with communicating Traffic Signal...
information, by a road-side-unit (RSU) to oncoming vehicles. Studies on improvement in traffic efficiency, through the implementation of GLOSA, has been carried out by [6], [16]. The aim of this work is not to test the efficacy of GLOSA but rather testing the capability of the V2X simulator being designed, to be capable of simulating V2X applications. The information conveyed via V2X, for GLOSA, would be the current phase, the phase remaining time and position of the traffic signal. This can be used to calculate the necessary speed that a vehicle should travel on in order to cross the traffic intersection without stopping. Code for GLOSA was written in MATLAB, which used information on packet drops/reception from NS3 in order to make velocity changes to appropriate vehicles in VISSIM.

4.1 Traffic Modelling

The traffic network model in VISSIM was based on the three intersection Alexandra Road (Fig 3a) corridor layout in Singapore. For the purpose of simulations and data collection, only one traffic signal head (with fixed-time signal control) on one of intersections is equipped with an RSU that is broadcasting the data packet and it is assumed that vehicles that are on the link served by the particular RSU can identify the relevant signal head, while other vehicles are capable of ignoring it. Also, a further assumption is that 100% of vehicle are equipped with OBUs. The intersection chosen was the second one from the left and V2X was tested on all vehicle travelling from West to East, through the afore-mentioned 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, shown in [3] which includes the afore-mentioned traffic intersection containing the V2X equipped signal head and the length of this section is approximately 650 m. Two sets of simulations were carried out, one with GLOSA and one without GLOSA. 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 an 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 EIRP is 40 dBi. Using this transmit power level ensured that GLOSA advisories could be communicated to vehicles at distances up to 600 metres. Vehicle numbers and their positions are arranged in an array in MATLAB and sent to NS3, while the network simulation 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, where T ∈ Z. The network simulation results will be communicated back to MATLAB which is 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 timing information conveyed by the traffic signal. The speed advisory is implemented at T+1, which incorporates a small delay in advisory assignment. This issue can be circumvented, if instead of providing accurate waypoints, we provide the current position and estimated future position as the two waypoints. However, since a vehicle path can change unexpectedly, the network simulation results obtained by using this mobility model may not be accurate [3]. Therefore, we decided carry out a trade-off for inclusion of accurate mobility model over delay in advisory assignment.
5 Results

Table 1 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. There is an obvious reduction in the values of all the performance parameters. 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. In addition, total carbon monoxide emissions per vehicle reduce from $171.4/99 = 1.73$ grams per vehicle to $196.2/122 = 1.608$ grams per vehicle and fuel consumption for traffic without GLOSA, which is approximately $14.5$ L/100 KM reduces to $13.4$ L/100 KM. There is a total decrease in fuel consumption by approximately 7.4%. 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.

<table>
<thead>
<tr>
<th></th>
<th>Total vehicles observed</th>
<th>Average vehicle delay</th>
<th>Average stopped delay</th>
<th>Average vehicle stops</th>
<th>Carbon Monoxide emissions</th>
<th>Fuel consumption</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>All vehicles (in Seconds)</td>
<td>Per Vehicle (in Seconds)</td>
<td>Per vehicle</td>
<td>Grams</td>
<td>Litres (All vehicles)</td>
</tr>
<tr>
<td>Without GLOSA</td>
<td>99</td>
<td>15.29</td>
<td>8.65</td>
<td>0.51</td>
<td>171.4</td>
<td>9.3</td>
</tr>
<tr>
<td>With GLOSA</td>
<td>122</td>
<td>15.08</td>
<td>3.69</td>
<td>0.25</td>
<td>196.2</td>
<td>10.62</td>
</tr>
</tbody>
</table>

Table 1: Simulation results

6 Conclusion and Future Work

The results shown above have been compared with the relevant results in [6], which reports a 7% reduction in fuel consumption and with [16], which reports a reduction of 50% in vehicle stopped delay, and has been found to be giving similar results. [6] uses SUMO and Jist/SWans while [16] does not incorporate a network simulator in their simulations. This aim of this work is to carry out a fundamental validation of the V2X simulator that is currently being developed. It has been envisioned to expand this simulator further in order to carry out simulations of a larger scale and also multiple V2X applications. One of our next steps, in this regards, would be to model and simulate 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 exploring the necessary Cooperative Adaptive Cruise Control strategies and how they can be realized if vehicles are able to exchange their state information via V2V.
References


