Parallel Closed-Loop Connected Vehicle Simulator for Large-Scale Transportation Network Management: Challenges, Issues, and Solution Approaches

Mohammad A. Hoque, *Senior Member, IEEE*, Xiaoyan Hong, *Member, IEEE*, Md Salman Ahmed, *Student Member, IEEE*

Abstract— **The augmented scale and complexity of urban transportation networks have drastically increased the execution time and resource requirements of vehicular network simulations, exceeding the capabilities of sequential simulators. The need for a parallel and distributed simulation environment is inevitable from a smart city perspective, especially when the entire city-wide information system is expected to be integrated with numerous services and ITS applications. In this paper, we present a conceptual model of an Integrated Distributed Connected Vehicle Simulator (IDCVS) that can emulate real-time traffic in a large metro area by incorporating hardware-in-the-loop simulation together with the closed-loop coupling of SUMO and OMNET++. We also discuss the challenges, issues, and solution approaches for implementing such a parallel closed-loop transportation network simulator by addressing transportation network partitioning problems, synchronization, and scalability issues. One unique feature of the envisioned integrated simulation tool is that it utilizes the vehicle traces collected through multiple roadway sensors—DSRC on-board unit, magnetometer, loop detector, and video detector. Another major feature of the proposed model is the incorporation of hybrid parallelism in both transportation and communication simulation platforms. We identify the challenges and issues involved in IDCVS to incorporate this multi-level parallelism. We also discuss the approaches for integrating hardware-in-the-loop simulation, addressing the steps involved in preprocessing sensor data, filtering, and extrapolating missing data, managing large real-time traffic data, and handling different data formats.**

Keywords—**Connected Vehicle, Parallel Simulation, Network Partitioning, Scalability, Communication Overhead, Dedicated Short Range Communication, hardware-in-the-loop simulation.**

I. Introduction

With the advent of big data and connected vehicle (CV) technologies, the parameters and requirements for simulating metro-scale urban transportation networks with heterogeneous vehicles have evolved substantially. Today's transportation engineers at the Traffic Management Centers (TMCs) feel the necessity of a parallel CV simulation tool that could allow them to visualize the immediate system-wide effect of any change in traffic parameters—signal timing, detour, lane closure—before making the decisions. Unfortunately, current state-of-the-art traffic simulators (VISSIM [1], CORSIM [2], SUMO [3] etc.) are designed and developed only for the microscopic simulation of vehicles and pedestrians, evaluation of traffic control algorithms, and visualization of on-road traffic behaviors. Since these traffic simulators cannot model and simulate the vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications (collectively known as V2X communications), additional work is necessary to port and feed the network simulators with these traffic simulators. Simulation of a transportation network with CV requires a bi-directional coupling mechanism between a transportation simulator and a communication simulator. This mechanism has led to the concept of the closed-loop CV simulator, which has recently drawn a significant amount of research interests within the community. However, the computational capacity of such a bi-directionally coupled (closed-loop) simulator is significantly limited by the number of CVs equipped with on-board units (OBUs) and the number of roadside units (RSUs) deployed within the metro-wide transportation network, since these DSRC devices transmit millions of basic safety messages (BSMs) packets every minute requiring massive computational resources. Existing sequential closed-loop simulators can barely handle one thousand vehicles simulated on a scenario involving no more than a few intersections. Thus, the incorporation of parallelism in both transportation and communication simulation platforms will enable efficient management of large-scale transportation network and control of traffic parameters involving connected vehicles. In addition, integrating roadway sensor data through hardware-in-the-loop simulation with the closed-loop software simulator will enable the traffic engineers to make informed decisions by evaluating the system-wide impact of traffic parameters changes in real-time.

A vast amount of research effort has been recently directed towards the improvement of surface transportation through self-driving autonomous vehicles as well as connected vehicles (CVs) using the 5.9 GHz Dedicated Short Range Communication (DSRC) technology. Automakers and technology developers like Google, Ford, and General Motors etc. are working to improve the controllability features of autonomous or semi-autonomous vehicles. While self-driving cars can potentially reduce the stress of navigating through congested traffic, CVs can optimize the traffic flow across an entire transportation network through the exchange of information among vehicles and infrastructure. CV applications use information obtained through V2X communications to assist drivers in avoiding congestion, reducing vehicle stops, choosing the best route, and optimizing fuel efficiency. Hence, CV-based emerging Intelligent Transportation Systems (ITS) applications can result in transformative changes to the overall surface transportation system.

To accurately simulate ITS applications on a scenario involving connected vehicles, it is necessary to integrate a full-fledged transportation simulator with a wireless network simulator, resulting in the need for a closed-loop simulator. This kind of closed-loop simulator requires a tight synchronization between two stand-alone simulation modules: a transportation module and a communication module. The transportation module is responsible for the modeling of vehicle mobility applications including traffic routing, car-following, lane-changing, vehicle dynamics, driver behavior modeling, and traffic signal control modeling etc. On the other hand, the communication module accounts for data traffic network modeling including packet routing, end-to-end V2X packet delivery, wireless media access, cross-layer protocols, information security, and authentication mechanisms.

In a CV simulation environment, the two simulation modules (transportation and communication) operate as a real-time feedback control loop with a tight synchronization. These two modules highly influence the operations of one another. For example, vehicle dynamics, mobility, speed, and density affect the communication links between vehicles as well as the data packet routing; hence, they also affect the communication quality, i.e., reliability, throughput, and delay. Conversely, the data communication parameters—for example, the number of packet losses between vehicles and the end-to-end delivery delay—can adversely affect the mobility decisions made by the transportation simulator, particularly when a V2X message carries detour information due to an accident. For a V2X-based safety application, it is important to realize that the slightest delay in communication, even about a fraction of a second, can have serious consequences and may even be fatal. Considering the complexity of the transportation and communication module with the high level of interdependency between them, it is easy to perceive how challenging the simulation of an integrated CV system can be.

II. Related Work

Most of the previous efforts to simulate vehicular networks were based on fixed mobility trajectories that were fed to the network simulator. Several mobility generator frameworks (VANETMOBISIM [7], SUMO [3], MOVE [8], STRAW [9], FREESIM [10], CITYMOB [11], C4R [38]) have been developed to produce the vehicular trajectories that are fed into various network simulators (NS2 [12], NS3 [13], OMNET++ [14], OPNET [15], JIST/SWANS [16], QualNET [17], etc.) without incorporating the effect of ITS applications on the mobility of the vehicles. Lee and Park [18] used the NCTUns communications simulator to examine the effects of communications using VISSIM trajectory data offline with no feedback loop for traffic simulation. GrooveSim [19] simulates inter-vehicular communication and vehicular mobility in a road traffic network using a customized mobility model and the GrooveNet [20] routing protocol. MobiREAL [21] incorporates mobility support on the Georgia Tech Network Simulator (GTNetS [22]). The capabilities of these type of open-loop simulators are limited to studying only unidirectional effects between the two domains. For example, studying the effect of various mobility models on the performance of end-to-end data communication using these simulators could characterize the dependency of the communication module on the transportation module, but it would be impossible to study the impact of data communication on the transportation system by incorporating changes in vehicle route, speed, signal timings, and mobility patterns based on newly received messages. Hence, this approach cannot be used to study bidirectional effects between the two tightly coupled domains.

Recently, there has been a significant amount of interests and efforts to design closed-loop CV simulators by coupling two types of simulators. Traffic and Network Simulation Environment (TraNS [23]) links the traffic simulator SUMO and the network simulator NS-2. Multiple Simulator Interlinking Environment for IVC (MSIE [24]) integrates NS-2, VISSIM traffic simulation, and application simulation (MATLAB) into a simulation environment for vehicular ad hoc networks (VANETs). Veins [4] provides a closed-loop integration using SUMO and OMNeT++ as the traffic and communication simulators respectively. Integrated Wireless and Traffic Platform for Real-Time Road Traffic Management Solutions (iTETRIS [25]) integrates SUMO with NS-3 through IP-based sockets and allows implementation of several ITS applications in various programming languages. Very recently, Songchitruksa et. al. developed a closed-loop CV simulator (CONVAS [5]) by interlinking VISSIM and NS-3. The project funded by the U.S. Federal Highway Administration's Exploratory Advance Research program initially targeted towards incorporating hardware-in-the-loop simulation [42] into their closed loop CV simulator CONVAS. However, as of now, no evidence of successful integration has been reported from that project. VNetIntSim [6], another recent closed-loop simulator, couples OPNET and INTEGRATION. A comprehensive survey of contemporary CV simulators can be found in one of our earlier papers [43]. We have summarized all these simulators in the Table 1:

Table 1: Summary of Integrated CV simulators

Integrated Simulator	*Traffic Simulator*	*Network module*	*Open-loop vs Closed-loop*
NCTUns [18]	VISSIM	NCTUnc	Open-loop
MOBIREAL [21]	CPE model	GTNeTS	Open-loop
GrooveSim [19]	Own model	GrooveNet	Open-loop
ASH [26]	IDM/MOBIL, IVG	SWANS	Closed-loop
OVNIS [27]	SUMO	NS-3	Closed-loop
Veins [20]	SUMO	OMNET++	Closed-loop
VnetIntSim [22]	INTEGRATION	OPNET	Closed-loop
TraNS [18]	SUMO	NS-2	Closed-loop
iTETRIS [21]	SUMO	NS-3	Closed-loop
CONVAS [23]	VISSIM	NS-3	Closed-loop

Unfortunately, none of these closed-loop simulators described above integrate with hardware-in-the-loop simulation technique. These tools also lack in providing support for simulating large-scale transportation scenario using parallel and distributed computing. Another major limitation is that there is no mechanism available for collecting roadway sensor data from individual intersections and feeding them to the simulation environment to facilitate real-time traffic decision support at the TMCs.

III. Parallel Simulation of Connected Vehicle Applications

A. *Need for Parallelism*

The augmented scale and complexity of urban transportation networks have drastically increased the execution time and resource requirements of vehicular network simulations, exceeding the capabilities of conventional simulators. The need for a parallel and distributed CV simulation environment is inevitable from a smart city perspective where the entire city-wide information system will be integrated with numerous services and ITS applications, particularly when the metro-wide multimodal transportation systems get connected to the smart city infrastructure through DSRC.

Unfortunately, contemporary simulation tools do not provide any mechanism for parallel or distributed simulation of CV applications for large-scale transportation networks. Due to the complexity of implementation, earlier attempts [46, 47] of parallelizing vehicular network simulation were limited to only open-loop simulators, which do not facilitate real-time feedback control between the transportation and communication parameters. To address the need for ITS practitioners and researchers, we develop a framework for parallelizing closed-loop CV simulation, with the option to incorporate real-time roadway sensor data through hardware-in-the-loop simulation.

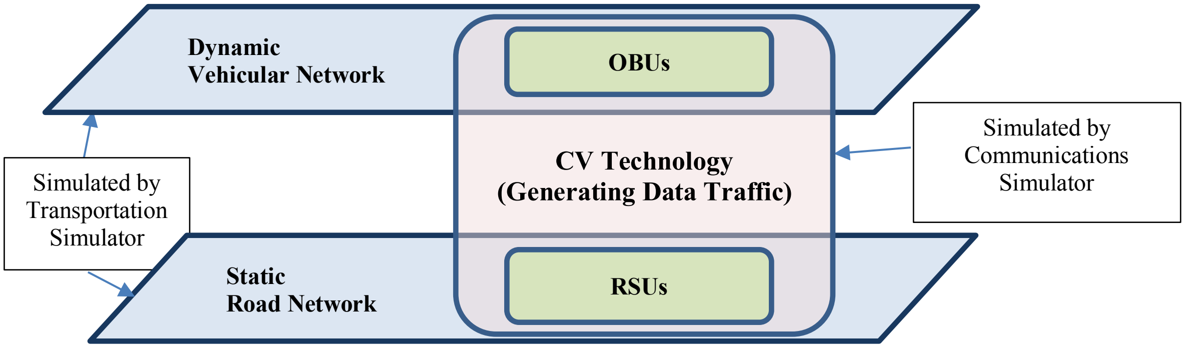


Figure 1: Bi-layer complex transportation network

B. *Implementation Challenges and Issues*

In this section, we identify some of the challenges and issues associated with implementing a parallel closed loop simulator for the large-scale transportation network management. Later we provide insights to the solution approaches that can address these problems.

1) *Partitioning of Bi-layer Complex Transportation Network*

The fundamental research problem involved in this parallel simulator design is to determine a near-optimal partitioning heuristic using a bi-layer network model—a static road network overlaid with a dynamic vehicular network—connected by the CV technology which spans across both the networks (Figure 1). Earlier research mainly focused on partitioning static road networks for distributed simulation without considering the data traffic generated by DSRC communications. The bi-layer model will address partitioning issues in both the dynamic vehicular network involving CVs (OBUs) as well as the static infrastructure (RSUs) and the interactions between the two levels. The solution approaches in section IV will provide guidelines to incorporate real-world roadway traffic parameters with the data traffic parameters within the partitioning heuristic for connected vehicle environments.

The biggest challenge in partitioning vehicular networks is that the partitions cannot be fully separated. In fact, due to the communication and high mobility, partitions have a high level of interdependency and interactivity (i.e., a message or a vehicle moves from one partition to another) that necessitate communications between partitions to achieve consistency and accuracy. Inefficient partitioning of such networks can produce high communication volume between partitions and high processing overhead within a partition, consequently resulting in low simulation speeds. So, it is necessary to create partitions in such a way that reduces the interactivity and interdependence between them. Another proven NP-hard problem is the load-balancing problem. Due to the interdependency between events in different partitions, the simulation must be synchronized between the partitions; i.e., low-load (high-speed) partitions must wait for high-load (low-speed) ones to finish. This means that the maximum overall simulation speed is limited to the minimum speed among all the partitions.

2) *Reducing Inter-Simulator Communication Overhead*

A major problem for parallelizing a closed-loop CV simulator is that it not only requires decomposing the two standalone simulators (the transportation and communication simulators) and synchronizing the components within each simulator, but it also requires a tight synchronization between the two simulators. DSRC technology requires that the vehicles broadcast their current locations every 0.1 seconds, meaning that the two simulators must synchronize ten times per second. This synchronization process adds extra overhead if the two simulators are running on separate computing nodes in a distributed computing environment requiring them to communicate over the Message Passing Interface (MPI). With a shared memory interface between the two simulators running in the same partition, this Inter-Simulator Communication overhead is expected to be reduced. However, using shared memory also creates a race condition between multiple processes running on the same computing node. Hence, there is always a trade-off between contention (shared memory) and latency (distributed memory), which is a major research problem. In addition, in a CV environment, the closed-loop interactions between communication and transportation systems must be executed in real-time to accurately model the impact of one system on its counterpart. For instance, the real-time interactions between SUMO and OMNET++ should facilitate dynamic speed control for the vehicles in the vicinity of traffic signals, where vehicles and signal controllers can exchange information to compute the optimal signal timing and vehicle trajectory.

3) Existence of heterogeneous vehicles

Another challenging aspect of simulating transportation network involving CVs is due to the slow market penetration rate of connected vehicles, which implies that during the transition period there will always be two types of vehicles on the road—one that is connected through DSRC (CV) and the other that is not connected (non-CV). It is expected that CV technologies will penetrate the market slowly over the next few years. Hence, until the time comes when all the cars on the road are equipped with factory-built or after-market DSRC devices, there will always be two types of vehicles on the road: one that has DSRC on-board unit (OBU) and the other that does not have OBU. CVs broadcast their actual GPS positions and speed every 0.1 seconds through the basic safety messages (BSMs). So, CVs can be easily identified through the BSM packets. However, detection of non-CV vehicles may not be easy, although it is possible to detect the non-CVs and reconstruct their mobility traces using several techniques. First, we can equip optical sensors to CVs so that the CVs can collect surrounding vehicles' information using the optical sensors. A CV can detect duplicate vehicles by communicating with other CVs. Second, we can reconstruct the mobility traces using some established techniques discussed in [39], [40], and [41]. Third, the data from the loop detector and traffic cameras can further refine non-CV detection and mobility trace reconstruction. At present, there is no closed loop simulator that supports the simulation of CV applications with heterogeneous types of vehicles.

Using the state-of-the-art closed-loop simulator with the support for hardware-in-the-loop simulation, vehicles with attached OBUs will be able to participate in the network-wide communication, while the vehicles without OBUs will not be detected by the simulator. Incorporating non-CVs in the hardware-in-the-loop simulation is quite challenging because the closed-loop CV simulator needs to be fed from several sources of sensor data such as CV traces through BSM messages and non-CV traces from roadway sensors (loop detectors and video detectors). From the input of these sources, the simulator needs to generate realistic mobility traces for the non-CVs, in addition to mapping the actual positions of the CVs where the simulator should graphically represent the CVs and non-CVs differently to distinguish between the actual position and speed vs. projected position and speed.

4) Synchronization problem

Simulation of data traffic is computationally more resource intensive than the simulation of vehicular traffic [6]. This makes the closed-loop simulation of CVs challenging because the imbalance of computational resource requirement causes synchronization problems between the transportation simulator and communication simulator. The synchronization problems happen due to the huge amount of DSRC basic safety messages (BSM) disseminated from each vehicle in every tenth of a second, where each BSM message needs to go through several layers of encapsulation and de-capsulation steps within the wireless network's protocol stack at both ends. Some of the services in the data communication protocols, e.g. error detection, routing, and connection establishment, are computationally more expensive compared to the services from the vehicular traffic simulator that do not require passing through multiple layers of protocols. In fact, simulation of vehicular traffic only involves trace generation using microscopic mobility models. Hence, the data traffic simulator primarily causes the bottleneck. Typically, the data traffic is simulated using network simulators such as OPNET, OMNET++, Qualnet, NS-2, or NS-3. One experiment in [29] demonstrated that the simulation of a 200-node network for only one minute generated more than 4,600,000 events and required 16 minutes of CPU time. The increasing complexity of the protocol stacks in end devices further aggravates this problem and has spurred efforts to develop parallel network simulators.

5) Scalability of Parallel Simulation

The scalability of parallel systems depends on the ratio of time spent in computation vs. communication. For any parallel system, the fraction of time spent in inter-process communication increases with the number of processors while the fraction of time spent in actual computation decreases. Initially, the computational time is greater than the communication time for a lower number of processors. The computational time decreases with respect to the communication time with the increase of the number of processors. At some point, for a specific number of processors (p), the communication time starts dominating over computation time. This value of p essentially determines how better the system is scalable. The higher the value of p, the better the scalability. Therefore, scalability is one of the most important problems in any parallel system, especially when it involves both distributed and shared memory architecture. Hence the architecture of such hybrid parallel system needs to be designed in such a way that reduces the inter-process communication (IPC) overhead and increases the scalability. It is noteworthy to mention that this IPC could take place between the transportation simulator and network simulator using shared memory (using OpenMP) or between the instances of the same simulator running different partitions on distributed cluster nodes (through MPI). Without achieving a certain level of scalability, the system will not be able to simulate a city-wide scenario with several hundred thousand vehicles and millions of BSM messages in every minute. Ahmed et. al. studied the scalability issues in terms of memory usage and execution time using VNetIntSim [33, 34] proving that the number of wireless nodes (vehicles) and the data traffic rate per vehicle are the primary reasons behind the scalability issue.

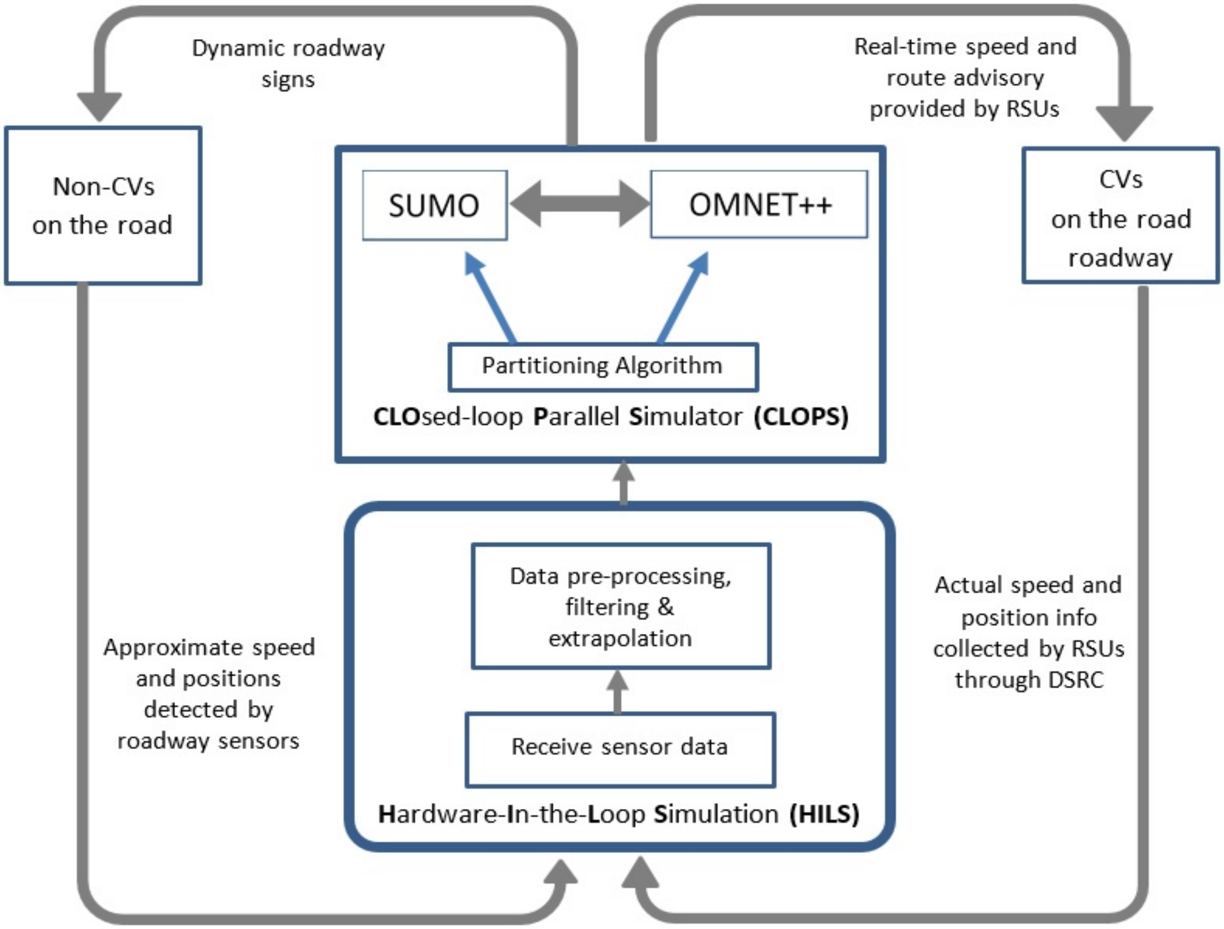

Dynamic roadway signs
Real-time speed and route advisory provided by RSUs
Non-CVs on the road
SUMO
OMNET++
CVs on the road roadway
Partitioning Algorithm
CLOsed-loop Parallel Simulator (CLOPS)
Data pre-processing, filtering & extrapolation
Receive sensor data
Hardware-In-the-Loop Simulation (HILS)
Approximate speed and positions detected by roadway sensors
Actual speed and position info collected by RSUs through DSRC

Figure 2: Conceptual Framework of Integrated Distributed CV Simulator (IDCVS)

IV. Conceptual Model

In this section, we present a conceptual framework of an **I**ntegrated **D**istributed **C**onnected **V**ehicle **S**imulator **(IDCVS)** and in section V we discuss the technical approaches for implementing such a robust simulation tool. Figure 2 below shows our hypothetical model of IDCVS system that includes hardware-in-the-loop simulation techniques for both CVs and non-CVs. IDCVS will incorporate two basic modules—a **CLO**sed-loop **P**arallel **S**imulator (**CLOPS**) and a **H**ardware-**I**n-the-**L**oop **S**imulation (**HILS**) module.

A. *Hardware-In-the-Loop Simulation (HILS)*

HILS will have an interface to receive the sensor data from both CVs and non-CVs through multiple sources. For non-CVs, the approximate location and speed can be detected through video detectors and inductive loop detectors, and this information will be passed as input to the HILS module. We can use the video detection and the loop detector software that can supply the sensor data to the HILS receiver component. On the other hand, the CV's can be detected easily through the BSM messages received by the RSUs. Once the sensor data is received, additional data-preprocessing, filtering, and extrapolation will be needed before the data can be used by CLOPS. This will require developing filtering algorithms for loop-detection and video-detection data to isolate the CV traces from the non-CV traces.

B. *CLOsed-loop Parallel Simulator (CLOPS)*

A CLOsed-loop Parallel Simulator (CLOPS) can be developed by interlinking OMNET++ with SUMO where both simulators are open source. An efficient partitioning heuristic will decompose the complex transportation network into two separate sets of partitions—where each set of the partition will be sent to the individual simulator (SUMO and OMNET++). It might appear that CLOPS could be developed as a parallel and distributed framework on top of Veins since Veins also utilizes a coupling between SUMO and OMNET++. However, since Veins does not support heterogeneous vehicles, it is not possible to extend Veins for the simulation scenarios involving both CVs and non-CVs. In addition, CLOPS may have the capability to vary the ratio of CVs to the non-CVs as per the market penetration rate. This requires a non-uniform partitioning between SUMO and OMNET++.

C. *Modes of Operation for IDCVS*

One important feature of this conceptual IDCVS system is that it will have the option to simulate in two different modes—closed-loop simulation (CLSim) mode and HILS-mode. The CLSim mode will simulate without sensor data, in this case the entire simulation will be run within CLOPS. To incorporate both DSRC-equipped and non-equipped vehicles on a CLSim scenario, we can randomly distribute the CV vehicles within the road network based on a user specified technology penetration rate. On the other hand, the HILS-mode will enable simulation based on real-time sensor data.

V. Implementation Approaches

In this section, we discuss the possible technical approaches to address the challenges pertaining to implementation of the integrated simulator.

D. *Developing Network Partitioning Heuristic*

A crucial challenge for the partitioning problem described in section III is that, due to the imbalance of computational resource requirements between transportation simulator (SUMO) and network simulator (OMNET++), a single partitioning scheme may not work for both othe simulators. Apart from that, the number of vehicles will also vary among the two simulators where SUMO needs to simulate the traces for all vehicles (both CVs and non-CVs) whereas OMNET++ only simulates data traffic generated from the CVs. If a single partitioning heuristic is used, the synchronization problem will be further aggravated. Hence, it is necessary to have two separate partitioning schemes for SUMO and OMNET++.

A Boolean matrix based vehicular network partitioning algorithm has been developed by Hoque et. al [44, 45] which is expected to be incorporated within our distributed simulation architecture involving hybrid parallelization model. In our recent work related to road-network partitioning [30, 31], we have identified the following issues and parameters that play vital roles in designing an efficient partitioning heuristic:

1) System boundary nodes of each partition: The total number of inter-process communication or messaging depends on the number of system boundary nodes of each partition.

2) The number of partitions: Almost every graph partitioning algorithm is based on a pre-specified number of partitions, which may not always generate the optimal solution in practice. Instead of specifying an exact number of partitions, an upper bound and lower bound can be provided as input to the algorithm to determine the best partitioning solution within the specified range.

3) Intersection cut: If an intersection is considered as a boundary node for a partition, then a significant amount of vehicle mobility data must be communicated between the partitions. In this context, an important factor—whether to prioritize signalized intersection over un-signalized intersection as a candidate for boundary node—remains open for further research, which should be investigated.

4) Link/Edge cut: When a link or edge is selected to be cut then the traffic volume along the cut link is directly proportional to the amount of information exchanged between the two partitions along the link. In this case, a good strategy would be to cut the links with minimum traffic to reduce the communication overhead between partitions.

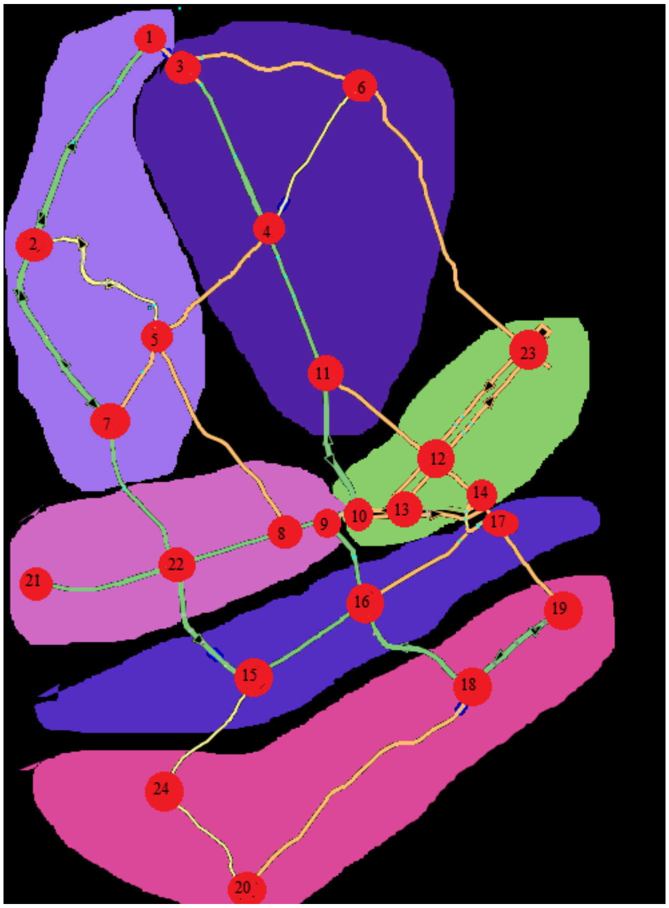


Figure 3: Partitions of the transportation corridor of Johnson City, Tennessee, United States

E. *Partitioning Approach for SUMO*

To create the network graph, the OSM file of the experimental city can be downloaded from the www.openstreet.org website. To avoid unnecessary complexities, residential street, service path, footway, cycleway, motorway, and unclassified roads can be excluded from the graph. Table 2 shows some suggested parameters that can be incorporated to generate the weighted graph matrix. It could be easily possible to extend an existing graph partitioning software like METIS [37] for generating the partitions of the transportation network. Figure 3 shows our preliminary transportation network partitioning using the above mentioned parameters with the help of METIS. METIS is a very stable partitioning package implementing the popular Kernighan-Lin heuristic. METIS divides a graph in three phases: coarsening, partitioning, and uncoarsening. In coarsening phase, the heavy edge matching scheme can be used, whereas, in the uncoarsening phase, the Kernighan-Lin graph refinement algorithm can be used. The coarsest graph can be bisected using graph growing followed by boundary Kernighan-Lin algorithm with graph partitioning using recursive bisection technique. The input for METIS can be provided using the generated graph matrix and weight parameters.

Table 2: List of parameters considered for partitioning heuristics

Parameter	Technique
Node weight	All signalized intersections in the OSM data will be identified. These types of intersections or nodes will be assigned a higher weight. Un-signalized intersections will be assigned the sum of the number of incoming and outgoing lanes as the weight.
Link length	The length between two nodes will be calculated using the Haversine formula: $d = 2r\,\sin^{-1}\left(\sqrt{\sin^2\left(\frac{\varphi_2-\varphi_1}{2}\right) + \cos\,(\varphi_1)\cos\,(\varphi_2)\sin^2\left(\frac{\lambda_2-\lambda_1}{2}\right)}\right)$ where d=Distance between two points/nodes, r=Radius of Earth (6367 km) $\varphi_{1,}\varphi_2$=Latitude of point 1 and 2 and λ_1, λ_2= Longitude of point 1 and 2
Number of lanes	The number of lanes of a road segment or a link will be extracted from the OSM data.
Link density	The density of a road segment or link will be extracted from the Google Map Application's newly introduced traffic layer [36]. The density is expressed in three categories: low, medium, and high.
Link priority	The road segment will be assigned a priority index based on the weighted summation of link length, the number of lanes, and link density.

F. *Partitioning Approach for OMNET++*

Some of the key factors concerning partitions for OMNET++ in the context of closed-loop parallel simulation for reduced interactivity and interdependence include vehicle mobility, communication events and external stimuli from the simulated transportation applications. These factors directly influence the previously mentioned challenges pertaining to network partitioning. The application stimuli are the drive for CV communications, which can be sporadic or proactive. The transportation network information such as the road network (road links, road nodes), vehicle density on each link, and vehicle speeds and distribution determine the vehicular mobility. This information can be further utilized to quantify the number of communication events. The approach to optimize partitions is to consider the number of discrete events in the communication network as the basis for drawing the boundary between the connected components. For example, one way to incorporate this approach is to employ the vehicle density and the length of each link as link weights in partitioning techniques (such as the minimum cut or minimum k-cut algorithms) to partition the network and minimize the interactivity between different portions. The lower the density and the longer the length of a link, the higher the possibility that the link is a cut link in the network. The rationale is that the density and length represent the continuity of the communication route on this link. Therefore, the lower this ratio (density/length), the less communication between the ends of the link. In addition, the partitions need to be adaptive to the dynamics of the application stimuli and the mobility. To address this issue, we can consider the simulation granularity and duration of the current partition time. The goal is to develop an intelligent algorithm to schedule the partitioning job.

G. *Design of Closed-loop Parallel Simulator (CLOPS)*

The closed-loop parallel simulator (CLOPS) integrates SUMO and OMNET++ as two standalone simulators. OMNET++ has the flexibility to dynamically create and delete nodes; this capability is necessary for a parallel simulation environment since the wireless vehicular nodes will be distributed in multiple network partitions based on geographic location. In addition, OMNET++ provides support for both distributed and shared memory computing which is needed for this project. The PHY and MAC layers of DSRC (IEEE 802.11p and IEEE 1609.4) have already been implemented in the OMNET++ platform by the open-source research community, which can be utilized in our research. This is a big advantage compared with OPNET since OPNET does not currently include the DSRC protocol stack.

CLOPS will incorporate hybrid parallelization schemes for both the traffic simulator and network simulator. This hybrid parallelization schemes will allow the integrated platform to run in parallel on clusters of computers within a supercomputing facility. The hybrid inter-process communication will be incorporated using MPI and OpenMP. In our preliminary work in [32], we investigated the impact of communication overhead on the overall simulation time and implemented a new partial-result accumulation pattern for reducing inter-process communication overhead. Figure 4 illustrates the architecture of the envisioned parallel system that incorporates hybrid parallelism. Both transportation and communication simulators will have master controllers (the Transportation Simulation Controller (TSC) and Network Simulation Controller (NSC)) that will coordinate the computational load distribution among the parallel sub-processes. Each of these sub-processes is supposed to simulate a portion of the transportation network defined by the network partitioning scheme. The controller will communicate with the sub-processes using MPI, while a transportation simulator sub-process corresponding to a specific partition communicates with its network simulator counterpart using OpenMP.

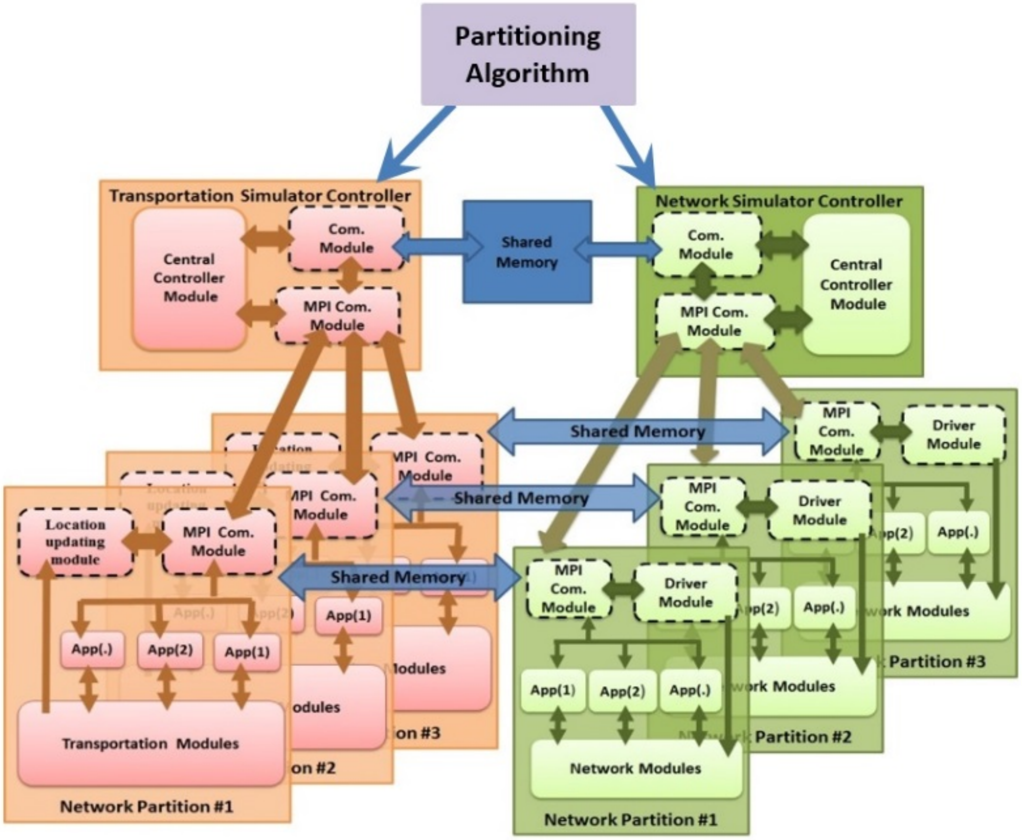


Figure 4. Envisioned distributed simulation architecture incorporating hybrid parallelism

It would be beneficial to utilize two levels of parallelization: network and event levels. At the network level, the overall network can be divided into multiple partitions for both SUMO and OMNET++, each of which will run on a different machine. The TSC and NSC are responsible for managing the loads and synchronizing the partitions within the transportation and communication domains, respectively. At the event level, events can run in parallel within a pre-calculated look-ahead interval. The calculation of the optimum look-ahead interval is crucial in the event-level parallelization. In fact, the look-ahead interval involves a tradeoff between the simulation speed and output accuracy. In the event-level parallelization, utilizing parameters such as node locations and the number of hops between two nodes can increase the scalability of the parallel simulation. For instance, nodes that are spatially separated by long distances can run events in parallel within longer look-ahead intervals without affecting the output.

The communication between the TSC and NSC can be achieved by using shared memory. The vehicles' locations will be calculated and sent to the NSC periodically through the shared memory, and any required application information between the TSC and NSC will be exchanged through the shared memory. Compared to TCP/IP message passing, shared memory has the advantages of reliability and the highest possible speed of information exchange. In contrast, the message size in TCP/IP message passing is limited; thus, in the case of large network size, a large number of messages are needed for each location update. Consequently, TCP/IP message passing may create a communication bottleneck, resulting in the degradation of simulation speed.

H. *Incorporating Hardware-In-the-Loop Simulation (HILS)*

The benefits of incorporating hardware-in-the-loop simulation in the parallel simulation framework is twofold: (a) real-time visualization of system-wide traffic and (b) fidelity testing of V2I applications. They are described as follows.

a) Visualization:

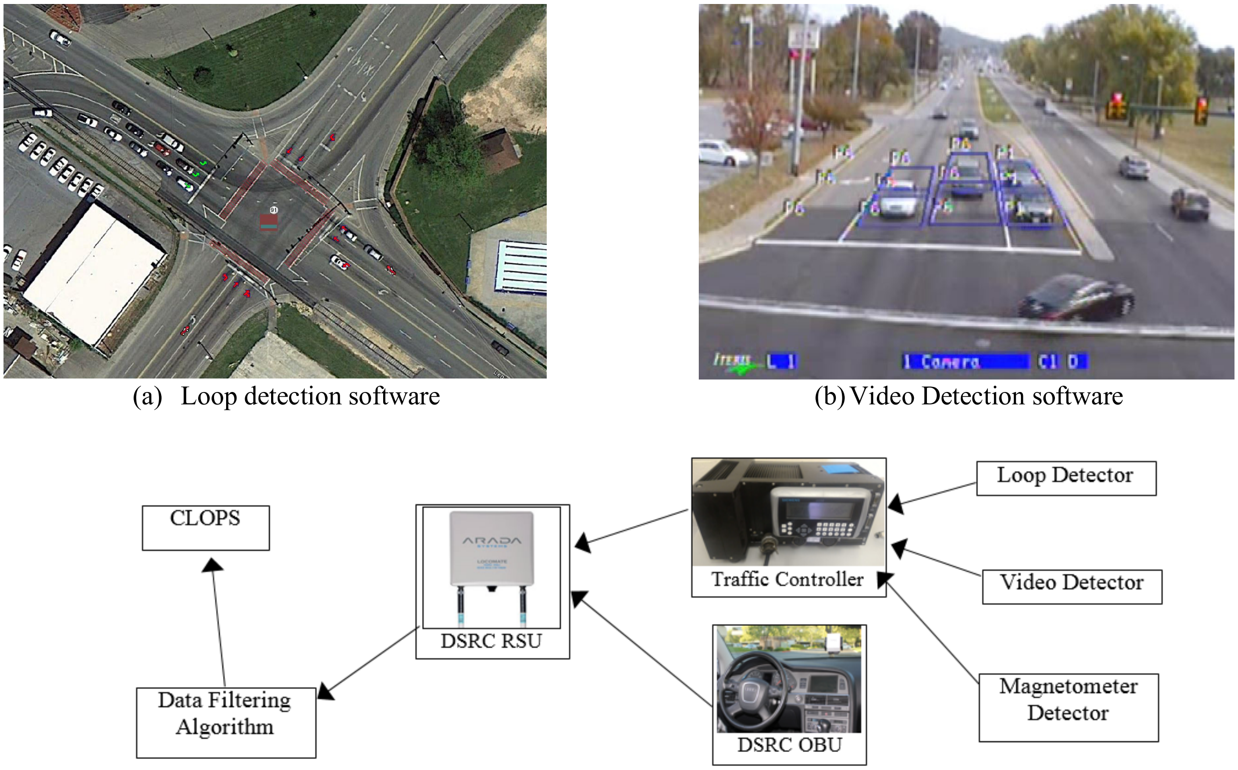
One of the purposes of integrating hardware-in-the-simulation (HILS) is to provide real-time traffic information obtained through various roadway sensors to the ITS practitioners monitoring roadway conditions from TMC (traffic management center). The HILS module aggregates roadway sensor data to facilitate visualization of system-wide traffic. A potential use case for our HILS-incorporated parallel CV simulation tool is to provide realistic prediction of the consequences of traffic change—such as transit bus delays or tentative queue length considering the preemptive detour advisory disseminated through DSRC—enabling a TMC official to make informed decision when a major corridor needs to undergo closure of lanes due to maintenance.

b) Fidelity Testing:

Another important benefit of integrating HILS is to validate the simulation results of V2I applications. For example, eco intersection approach applications [48-60] using signal phase and timing (SPaT) information could be first simulated in the CLOsed-loop Parallel Simulator (CLOPS) and then validated using actual roadway sensor data through HILS.

To capture the movement of the non-CVs, several types of detectors can be used such as the magnetometer, inductive loop detection (ILD), video detection, etc. Loop detection is also capable of counting traffic. But it is not 100% reliable for actual traffic counts because the loops in the adjacent through lanes are often tied together for one output for the movement. To solve this problem, the latest video detection technology capable of counting actual traffic can act as a complement for the loop detector. Figures 5(a) and 5(b) shows how the two software detect vehicles at the intersection through software.

Since the target is to simulate both CVs and non-CVs, it is necessary to feed the vehicles' information to the traffic and communication simulators. The RSU can automatically detect the CVs from the BSM packets, but the loop detection and video detection techniques are necessary for detecting the non-CVs. Once the RSU gets the data from all the sources (e.g. BSM packets, inductive loop, video, and magnetometer), a filtering algorithm separates the non-CVs from the CVs using the BSM packets. There is a re-identification issue regarding the use of loop detector, video detection, and magnetometer. In order to eliminate the re-identification problem, we will consolidate the roadway sensor data from three sources along with CV data to accurately identify both CVs and non-CVs. However, detection of the non-CV is not sufficient for the hardware in loop simulation. The mobility trace of a non-CV between two intersections is needed. A car-following model between one/two CVs and a non-CV can be used to extrapolate the missing trace of a non-CV vehicle. For example, the missing mobility trace of a non-CV vehicle can be extrapolated using two CVs' mobility traces where one CV precedes the non-CV and one CV that follows the non-CV. More generally, we can utilize the existing stochastic or probabilistic traffic flow and mobility trace reconstruction techniques [39 - 41] to reconstruct the mobility traces of the non-CV vehicles considering the CV vehicles as the probe vehicles (details discussed in the second challenge of HILS). Other sensors in an intersection can aid to refine the reconstruction. Figure 5c shows the flow of sensor data for hardware-in-the-loop simulation.



(a) Loop detection software

(b) Video Detection software

(c) Flow of data between sensors

Figure 5. Integrating hardware-in-the-loop simulation techniques

Some of the challenges associated with integrating hardware-in-the-loop simulation are described below:

1) Isolating CV traces from loop-detection and video detection data

The hardware-in-the-loop simulation (HILS) technique can capture roadway sensor data from four different sources—DSRC broadcast messages, inductive loops, video detectors, and wireless magnetometers. Unfortunately, the roadway sensors at intersections cannot differentiate between a CV and non-CV. So, a reliable filtering mechanism is needed to identify the CVs among all the traffic by filtering out the CV data from other two sensors' data based on the GPS position and loop detection timings.

2) Missing traces between two intersections

It is very challenging to emulate non-CVs based on sensor data because of the missing traces between two intersections since they can only be detected at the intersections. Also, SUMO generated mobility traces between two intersections are the only sources to fill up the missing traces. However, this approach may give some margin of error since some vehicles may arrive at their destination before reaching the next intersection while some other vehicles may start from a mid-point between the two intersections. Since the goal is to approximate the expected traffic between two intersections at a given time, some established statistical models are necessary to validate the simulation results between two intersections. For example, the modal activity-based probabilistic model [39] can be utilized to reconstruct missing non-CV traces using the CV traces. The modal activity sequence in the probabilistic model [39] utilizes the repetitive stop-and-go behaviors of vehicles between two intersections which in fact result a particular pattern of the activities such as idling, acceleration, cruising, and deceleration. The model generates several vehicle dynamic states using the pattern, ranks the dynamic states using probability distribution functions, and reconstructs the second by second trajectories from the dynamic states. It might be beneficial to incorporate the traffic flow reconstruction methods described in [40] and [41] where each method discusses the traffic flow reconstruction technique using the traffic data such as speeds, positions, and timestamps collected by probe vehicles. In our model, we plan to reconstruct the traces of non-CV vehicles considering the CV vehicles as the probe vehicles. The trajectories of CVs will be reflected in the simulation with almost 100% accuracy given the precision level of GPS sensors integrated with DSRC OBUs. On the other hand, the trajectories of the non-CVs can be deterministically traced near the intersections with the aid of roadway sensors. This means that the non-CVs (real) will be transformed into virtual vehicles between two intersections, while CVs will continue to provide actual mobility traces even within the road segment between two intersections. Between two intersections, the traces of non-CVs can be reconstructed up to a certain level of accuracy by utilizing relative distance information obtained through the sensors equipped in CVs. Even if there is no CV available within the vicinity of a virtual non-CV, still the recent microscopic car-following models can generate a 'close-to-real' trajectory between two intersections in SUMO.

3) *Inaccurate traffic count by loop detectors*

Loop detection can detect traffic but is less reliable for actual traffic counts because the loops in the adjacent through lanes are often tied together for one output for the movement. Also, due to the length of the loop (40 to 50 feet) at the stop bar, multiple vehicles may fall over the same loop or the loops tied together at the same time which reduces vehicle count accuracy.

The current reported accuracy from trace reconstruction model is about 80% [40, 41]. We believe that once we consolidate the sensor data from three different sources and cross match with CV data, the accuracy will be increased. However, some complex vehicle identification algorithm has to be developed to accomplish this goal. The improvement of accuracy level from roadway sensor data is left as an open research issue for the community.

4) *Different data formats*

Typically, data loggers' records include events at an intersection, including a light turning green, a light turning yellow, a vehicle detector turning on, a vehicle detector turning off, and pedestrian walk phase active. While CV data follows DSRC beacon format, loop detector, and video detector inputs are again in a different format. Thus, different pre-processing algorithms are needed.

CONCLUSION

In this paper, we have discussed a conceptual model that can simulate system-wide changes in traffic parameters on roadways involving both connected vehicles and regular vehicles. We have identified the major challenges and issues for implementing the hardware-in-the-loop simulation and incorporating hybrid parallelism in the closed-loop simulation. We have also discussed the solution approaches for the challenges and issues involved in implementing the conceptual model. However, only a few solutions have actually been implemented. We have discussed possible technical approaches to address the challenges and implementation issues. Our ongoing efforts are directed towards implementation of this model and evaluation of the scalability for emulating metro-wide transportation network. It is noteworthy to mention that a closed-loop integrated parallel simulator may still lack some realistic features within individual simulator modules (e.g. SUMO and OMNET++). However, given that both of the standalone simulators (SUMO and OMNET++) are open-source in nature, it is possible to add the missing features or refine the parameters within each of the simulators (SUMO and OMNET++) even after the implementation of closed-loop simulator. Our focus in this paper is to point out the challenges of designing the interface between two simulators in parallel platform, considering the traffic simulator and network simulator as a "black box".

REFERENCES

1. Fellendorf, Martin. "VISSIM: A microscopic simulation tool to evaluate actuated signal control including bus priority." 64th Institute of Transportation Engineers Annual Meeting. Springer, 1994.
2. Halati, Abolhassan, Henry Lieu, and Susan Walker. "CORSIM-corridor traffic simulation model." Traffic Congestion and Traffic Safety in the 21st Century: Challenges, Innovations, and Opportunities. 1997.
3. Krajzewicz D, Rossel C. Simulation of Urban MObility (SUMO). German Aerospace Centre, 2007. Available at: http://sumo.sourceforge.net. Last Accessed: 14 November 2016.
4. Sommer, C., R. German, and F. Dressler. Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis. IEEE Transactions on Mobile Computing, Vol. 10, No. 1, 2011, pp. 3-15.
5. Praprut Songchitruksa, Srinivasa Sunkari, Ines Ugalde, Juan Aparicio Ojea, and Justinian Rosca. Integrating Vissim and ns-3 for Connected/Automated Vehicle Simulation: A Case Study of Intelligent Dilemma Zone Avoidance, Proceedings of the ITS America Annual Meeting, 2016.
6. Ahmed Elbery, Hesham Rakha, Mustafa Y. ElNainay and Mohammad A. Hoque, VNetIntSim - An Integrated Simulation Platform to Model Transportation and Communication Networks. Proceedings of International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS), 2015.
7. Haerri J, Fiore M, Fethi F, Bonnet C. VanetMobiSim: generating realistic mobility patterns for VANETs. Institut Eurécom and Politecnico Di Torino, 2006. Available at: http://vanet.eurecom.fr/. Last Accessed: 14 November 2016.
8. MOVE (MObility model generator for VEhicular networks): Rapid Generation of Realistic Simulation for VANET, 2007. Available at: http://lens1.csie.ncku.edu.tw/MOVE/index.htm. Last Accessed: 14 November 2016.
9. STRAW - STreet RAndom Waypoint - vehiclar mobility model for network simulations (e.g., car networks), 2008. Available at: http://www.aqualab.cs.northwestern.edu/projects/144-straw-street-random-waypoint-vehicular-mobility-model-for-network-simulations-e-g-car-networks, Last Accessed: 14 November 2016.
10. FreeSim, 2008. Available at: http://www.freewaysimulator.com/
11. Martinez FJ, Cano JC, Calafate CT, Manzoni P. Citymob: a mobility model pattern generator for VANETs. In IEEE Vehicular Networks and

Applications Workshop (Vehi-Mobi, held with ICC), Beijing, China, May 2008.

12. Ns-2 Network Simulator. http://nsnam.sourceforge.net/wiki/index.php/Main_Page , Accessed November 11, 2016.
13. ns-3 Network Simulator. https://www.nsnam.org/ , Accessed November 11, 2016.
14. A. Varga, "The OMNeT++ discrete event simulation system," Proceedings of the European simulation multiconference (ESM'2001), vol. 9, p. 65, 2001.
15. Riverbed Technologies, 2008. Available at: https://www.riverbed.com/products/steelcentral/opnet.html, Last Accessed: 14 November 2016.
16. JiST/SWANS: Java in Simulation Time/Scalable Wireless Ad hoc Network Simulator, 2004. Available at: http://jist.ece. cornell.edu/
17. Scalable Network Technologies. Qualnet. Scalable Network Technologies, Inc., 2006. Available at: http://web.scalable-networks.com/qualnet-network-simulator, Last Accessed: 14 November 2016.
18. Lee, J., and B. B. Park. Investigating Communications Performance for Automated Vehicle-Based Intersection Control under Connected Vehicle Environment. IEEE Intelligent Vehicles Symposium (IV), Seoul, Korea, June 28 - July 1, 2015.
19. Mangharam R, Weller D, Rajkumar R, Mudalige P, Bai F. GrooveNet: A Hybrid Simulator for Vehicle-to-Vehicle Networks. Carnegie Mellon University, 2006. Available at: https://github.com/mlab-upenn/GrooveNet, Last Accessed: 14 November 2016.
20. R. Mangharam, D. S. Weller, R. Rajkumar, Priyantha Mudalige and Fan Bai, "GrooveNet: A Hybrid Simulator for Vehicle-to-Vehicle Networks", Second International Workshop on Vehicle-to-Vehicle Communications (V2VCOM), San Jose, USA. July 2006.
21. MobiREAL, 2008. Available at: http://www.mobireal.net/, Last Accessed: 14 November 2016.
22. The Georgia Tech Network Simulator (GTNetS), 2008. Available at: http://www.ece.gatech.edu/research/labs/MANIACS/GTNetS/, Last Accessed: 14 November 2016.
23. Piorkowski, M., M. Raya, A. 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, 2008, pp. 31-33.
24. Lochert, C., A. Barthels, A. Cervantes, and M. Mauve. Multiple Simulator Interlinking Environment for IVC. 2nd ACM International Workshop on Vehicular Ad Hoc Networks (VANET 2005), Cologne, Germany, September 2, 2005, pp. 87-88.
25. Rondinone, M., J. Maneros, D. Krajzewicz, R. Bauza, P. Cataldi, F. Hrizi, J. Gozlvez, V. Kumar, M. Rockl, L. Lin, O. Lazaro, J. Leguay, J. Haerri, S. Vaz, Y. L. M. Sepulcre, M. Wetterwald, R. Blokpoel, and F. Cartolano. iTETRIS: A Modular Simulation Platform for the Large Scale Evaluation of Cooperative ITS Applications. Simulation Modelling Practice and Theory, Vol. 34, Elsevier, 2013.
26. K. a. W. M. C. Ibrahim, "ASH: Application-aware SWANS with highway mobility," INFOCOM Workshops 2008, IEEE, pp. 1-6, 2008.
27. Y. a. D. G. a. B. P. Pigné, "A platform for realistic online vehicular network management," GLOBECOM Workshops (GC Wkshps), 2010 IEEE, pp. 595-599, 2010.
28. U.S. Department of Transportation, https://www.transportation.gov, Last Accessed: 13 November 2016.
29. W. Hao, R. M. Fujimoto, and G. Riley, "Experiences parallelizing a commercial network simulator," in *Simulation Conference, 2001. Proceedings of the Winter*, 2001, pp. 1353-1360 vol.2.
30. Ahmed, M.S & Hoque, M., "Partitioning of Urban Transportation Networks Utilizing Real-World Traffic Parameters for Distributed Simulation in SUMO," IEEE Vehicular Network Conference, Columbus, Ohio, 2016.
31. Ahmed, M.S & Hoque, M., "Partitioning of Urban Transportation Networks Using Evolutionary Algorithm for Distributed Simulation in SUMO," ACM-Mid Southeast Conference, Gatlinburg, Tennessee, 2016.
32. Ahmed, M.S.; Houser, J.; Hoque, M. & Pfeiffer, P., "Reducing Inter-process Communication Overhead in Parallel Sparse Matrix-Matrix Multiplication," International Journal of Grid and High Performance Computing (IJGHPC), March, 2017.
33. Ahmed Elbary, Hesham Rakha, Mustafa ElNainay, Mohammad A Hoque, "An Integrated Architecture for Simulation and Modeling of Small- and Medium-Sized Transportation and Communication Networks", Smart Cities, Green Technologies, and Intelligent Transport Systems, Springer Lecture notes in Communications in Computer and Information Science, ISBN: 9783319277530
34. Ahmed Elbary, Hesham Rakha, Mustafa ElNainay, Mohammad A Hoque, "VNetIntSim: An Integrated Simulation Platform to Model Transportation and Communication Networks," International Conference on Vehicle Technology and Intelligent Transport Systems 2015.
35. Do Not Pass Warning Application. Available at: http://www.its.dot.gov/infographs/DoNotPass.htm (*Accessed Nov 13, 2016*)
36. https://productforums.google.com/forum/#!topic/maps/byNxu_lT0do
37. Karypis, George, and Vipin Kumar. "METIS--unstructured graph partitioning and sparse matrix ordering system, version 2.0." (1995).
38. M. Fogue, P. Garrido, F. J. Martinez, J. C. Cano, C. T. Calafate, and P. Manzoni. "A realistic simulation framework for vehicular networks." Proceedings of the 5th International ICST Conference on Simulation Tools and Techniques (SIMUTools), Desenzano, Italy, 19-23 march 2012.
39. Hao, Peng, et al. "Modal Activity-Based Stochastic Model for Estimating Vehicle Trajectories from Sparse Mobile Sensor Data." IEEE Transactions on Intelligent Transportation Systems 18.3 (2017): 701-711.
40. Ramezani, Mohsen, and Nikolas Geroliminis. "On the estimation of arterial route travel time distribution with Markov chains." Transportation Research Part B: Methodological 46.10 (2012): 1576-1590.
41. Zheng, Fangfang, and Henk Van Zuylen. "Urban link travel time estimation based on sparse probe vehicle data." Transportation Research Part C: Emerging Technologies 31 (2013): 145-157.
42. https://tti.tamu.edu/2014/06/01/testing-connected-transportation-innovations-starts-with-first-creating-the-test-itself/
43. M. S. Ahmed, M. A. Hoque and P. Pfeiffer, "Comparative study of connected vehicle simulators," SoutheastCon 2016, Norfolk, VA, 2016, pp. 1-7.
44. Hoque, Mohammad A., Xiaoyan Hong, and Brandon Dixon. "Efficient multi-hop connectivity analysis in urban vehicular networks." Vehicular Communications 1.2 (2014): 78-90.
45. Hoque, Mohammad Asadul, Xiaoyan Hong, and Brandon Dixon. "Analysis of mobility patterns for urban taxi cabs." Computing, Networking and Communications (ICNC), 2012 International Conference on. IEEE, 2012.
46. Lee, Der-Horng, and P. Chandrasekar. "A framework for parallel traffic simulation using multiple instancing of a simulation program." ITS Journal 7, no. 3-4 (2002): 279-294.
47. Luciano Bononi, Marco Di Felice, Marco Bertini, and Emidio Croci. 2006. Parallel and distributed simulation of wireless vehicular ad hoc networks. In Proceedings of the 9th ACM international symposium on Modeling analysis and simulation of wireless and mobile systems (MSWiM '06). ACM, New York, NY, USA, 28-35.
48. Applications for the Environment: Real-Time Information Synthesis (AERIS) Program: AERIS Research Program Overview. URL: https://www.its.dot.gov/research_archives/aeris/index.htm
49. Xia, Haitao, Kanok Boriboonsomsin, Friedrich Schweizer, Andreas Winckler, Kun Zhou, Wei-Bin Zhang, and Matthew Barth. "Field operational testing of eco-approach technology at a fixed-time signalized intersection." In Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on, pp. 188-193. IEEE, 2012.
50. Xia, Haitao. Eco-approach and departure techniques for connected vehicles at signalized traffic intersections. University of California, Riverside, 2014.
51. Hao, Peng, Guoyuan Wu, Kanok Boriboonsomsin, and Matthew J. Barth. "Developing a framework of eco-approach and departure application for actuated signal control." In Intelligent Vehicles Symposium (IV), 2015 IEEE, pp. 796-801. IEEE, 2015.
52. Li, Weixia, Guoyuan Wu, Matthew J. Barth, and Yi Zhang. "Safety, mobility and environmental sustainability of Eco-Approach and Departure application at signalized intersections: A simulation study." In Intelligent Vehicles Symposium (IV), 2016 IEEE, pp. 1109-1114. IEEE, 2016.
53. Zweck, Michael, and Michael Schuch. "Traffic light assistant: Applying cooperative ITS in European cities and vehicles." In Connected Vehicles and Expo (ICCVE), 2013 International Conference on, pp. 509-513. IEEE, 2013.

54. Xia, Haitao, Guoyuan Wu, Kanok Boriboonsomsin, and Matthew J. Barth. "Development and evaluation of an enhanced eco-approach traffic signal application for connected vehicles." In Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on, pp. 296-301. IEEE, 2013.
55. Cho, Cheng-Hsuan, Han Su, Yi-Hong Chu, Wen-Yao Chang, and Frank Chee-Da Tsai. "Smart moving: A SPaT-based advanced driving-assistance system." In Network Operations and Management Symposium (APNOMS), 2012 14th Asia-Pacific, pp. 1-7. IEEE, 2012.
56. Zhao, Yiran, Shen Li, Shaohan Hu, Lu Su, Shuochao Yao, Huajie Shao, Hongwei Wang, and Tarek Abdelzaher. "Greendrive: A smartphone-based intelligent speed adaptation system with real-time traffic signal prediction." In Proceedings of the 8th International Conference on Cyber-Physical Systems, pp. 229-238. ACM, 2017.
57. Koukoumidis, Emmanouil, Li-Shiuan Peh, and Margaret Rose Martonosi. "SignalGuru: leveraging mobile phones for collaborative traffic signal schedule advisory." In Proceedings of the 9th international conference on Mobile systems, applications, and services, pp. 127-140. ACM, 2011.
58. Zhao, Yiran, Yang Zhang, Tuo Yu, Tianyuan Liu, Xinbing Wang, Xiaohua Tian, and Xue Liu. "CityDrive: A map-generating and speed-optimizing driving system." In INFOCOM, 2014 Proceedings IEEE, pp. 1986-1994. IEEE, 2014.
59. Alexander, W., X. Hong, and A. Hainen. "V2I Communication-Enabled Real-Time Intersection Traffic Signal Scheduling." In Proceedings of the SouthEast Conference, pp. 26-33. ACM, 2017.
60. Li, Xiaobing, Asad J. Khattak, and Airton G. Kohls. "Signal phase timing impact on traffic delay and queue length-a intersection case study." In Winter Simulation Conference (WSC), 2016, pp. 3722-3723. IEEE, 2016.