Towards Real-Time Interactive V2X Simulation

Dominik S. Buse and Falko Dressler
Heinz Nixdorf Institute and Dept. of Computer Science, Paderborn University, Germany
{buse,dressler}@ccs-labs.org

***Abstract*—Vehicle-to-Everything (V2X) communication technology is supposed to turn separate vehicles into a connected system of road users in the foreseeable future. To develop and test such systems in a holistic fashion today, the options are to use either large-scale network simulation or small-scale experimentation. Our ego vehicle approach tackles this problem by coupling existing tools of different scales and levels of detail. However, the coupling of real-time systems with non-real-time event-based simulation brings many new challenges. We propose to simulate only a selection of road traffic and communication around a selected ego vehicle, to reduce the computation effort to an achievable amount. We showcase our approach for an urban scenario with beaconing using the Veins simulator controlled via Ego Vehicle Interface (EVI). Our results show that only a small number of vehicles are needed to perform a complete simulation for the perspective of the ego vehicle in a typical urban scenario, which can easily be achieved on a typical PC platform. We see our real-time interactive V2X simulation platform as a step towards a more integrated way to design and test future connected cars.**

I. Introduction

Vehicle-to-Everything (V2X) technology will change the interactions of road traffic and connect the logic of its participants [1]. Vehicles are no longer just an isolated element in traffic, but a part of an interacting network. Cooperative awareness and deliberate coordination will enable applications beyond the limited concept of local reaction to sensor input. While this will bring many merits, it will also require a new way of designing and testing connected vehicles and their components. Test-benching individual components will no longer suffice as wireless networks will cause many components to influence each other across large distances [2]. Pure network-centric approaches cannot solve this issue, because they neglect the complexity of and decisions made by the intelligent cyber-physical systems that vehicles have become [3]. Also, their long-running nature will not help testing hardware prototypes that require real-time input. But there are only tools for these two approaches and implementing features of both in one tool fails due to resource limits: *Simulating a whole network of vehicles with the level of detail of a test bench or Hardware-in-the-Loop (HiL) simulator is not possible*.

But it may not always be necessary to simulate all vehicles with such a high level of detail. For the analysis of a specific vehicle and its behavior within an V2X environment, said environment could be simulated with a lower level of detail than the vehicle itself. The concept of the Ego Vehicle Interface (EVI) [4] harnesses this principle: A selected vehicle under test, the *ego vehicle*, is simulated in highest level of detail, e.g., by using HiL technology and/or prototypes. Alternatively, the ego vehicle is simulated in a driving simulator used by a human driver [5]. The environment of the ego vehicle is simulated in

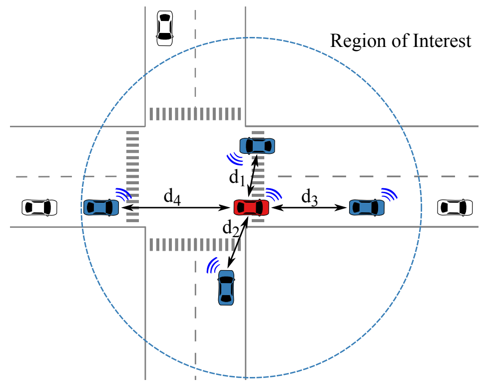


Figure 1. Dynamic floating ROI around the ego vehicle (in red, at the center). Communication is only simulated for vehicles within the ROI (in blue).

simulators with decreasing levels of detail.A V2X simulator, usually operating according to the event-based simulation principle, evaluates the wireless communication of vehicles in communication range, depending on the location of the ego vehicle. The rest of the vehicles in the scenario are simulated in an abstract fashion by a traffic simulator, supporting hundreds or thousands of vehicles. The EVI connects all these simulators and ensures timely exchange of vehicle information among the simulators. However, no guarantees for real-time operation of the coupled simulation approach can be given and the system relies on over-provisioning of compute power.

In this paper, we go one step further and thoroughly investigate the time bounds in the non-real-time simulation toolkit Veins [6], which is considered a typical example of V2X simulators. Based on this feasibility analysis, we derive an online algorithm for selecting a subset of cars in the proximity of the ego vehicle to be simulated in great level of detail by the V2X simulator (see Figure 1). All other cars only need to be considered for their mobility, which means details of the communication subsystem can be reduced substantially.

Our key contributions can be summarized as follows:

- We present a method to run real-time interactive V2X simulation focused on an ego vehicle (Section III),
- we derived a concept to validate the feasibility of focused real-time V2X simulation (Section IV-A),
- we developed a method to identify the amount of vehicles needs to be simulated (Section IV-B), and
- we experimentally show the real-time performance of the resulting system (Sections V and VI).

II. Related Work

Real-time feasible V2X simulation has been used for two major applications: prototype-testing and driving simulators.

Obermaier et al. [7] present a HiL system connected to the OMNeT++ network simulation framework via an actual wireless channel. A single Device under Test (DUT) is implemented in hardware while the other V2X-enabled vehicles are simulated. A similar setup was previously presented by Laux et al. [8]. Szendrei et al. [9] developed an orchestrator that uses the traffic simulator SUMO [10] to provide mobility to V2X HiL simulators. Wireless communication is not simulated but implemented using real radios. The authors claim to support 10 to 20 V2X systems, but do not provide more detailed analysis of their real-time feasibility.

Aramrattana et al. [11] present a driving simulator that is connected to Plexe, the platooning extension of Veins. Only a few vehicles are simulated in a scenario with no obstacle shadowing by buildings, which allows the system to run in real-time at an update rate of 100 Hz. However, no details on the exact number of vehicles and real-time feasibility are given. Egea-Lopez et al. [12] implemented a combined driving and traffic simulator with vehicle physics in Unity. Wireless signal propagation is simulated using a custom ray-shooting model running on a GPU. OMNeT++ and the INET framework can be connected to simulate communication protocols. The simulator is reported to support 150 vehicles in real-time. Zhao et al. [13] coupled a driving simulator with PARAMICS for traffic simulation and ns-2 for communication simulation. They used a fixed area of interest around an ego vehicle due to performance issues of communication simulator, even for a simple corridor scenario for the road network. However, they do not provide concrete numbers of vehicles.

Sliwa et al. [14] present a different coupling concept by fully integrating a V2X simulator into a single binary, LIMoSim, to avoid IPC for performance and usability reasons. While they do not analyze real-time feasibility directly, their measurements show that the system spends most of the computation time simulating wireless signals. And even for low numbers of vehicles, the computation time surpasses the simulated time when simulating WLAN-based communication.

To the best of out knowledge, no works so far have analyzed how many vehicles are needed for real-time V2X simulation and how many vehicles can be simulated by the simulator in real-time. Rather, the common approach so far is to either work on very few real-time components (which are often implemented in hardware) or use a number or region of vehicles that is supposedly large enough. However, the literature shows that the communication simulation seems to be the bottleneck for V2X simulation. With this paper, we address this issue and, building upon our Ego Vehicle Interface (EVI) [4], provide methods to determine the number of necessary and possible vehicles for real-time interactive V2X simulation.

III. Ego Vehicle Interface (EVI)

The goal of EVI [4] is to simulate the behavior and interactive surroundings of an ego vehicle in real-time. To achieve this, it couples simulation systems from different domains. These simulations systems are arranged in a hierarchy of different levels of detail. The ego vehicle (or even a specific component of it or its human driver) is at the center, the first layer of this hierarchy. It is simulated with the highest level of detail, but the smallest scale (i.e., only the ego vehicle itself). With increasing distance from the ego vehicle, the level of detail can easily become coarser, while the scale increases. The vehicles in the second level of the hierarchy are within sensor range of the ego vehicle and in direct interaction with it. They are simulated by a real-time system that depends on the system under test and can be a HiL simulator or a driving simulator. The vehicles in the third layer of the hierarchy are within communication range of the ego vehicle and exchanging V2X messages with it. They are simulated by the V2X simulator Veins [6]. The vehicles in the fourth and outermost layer of the hierarchy are simulated by the traffic simulator SUMO [10]. All vehicles in higher layers are also present in the layers below.

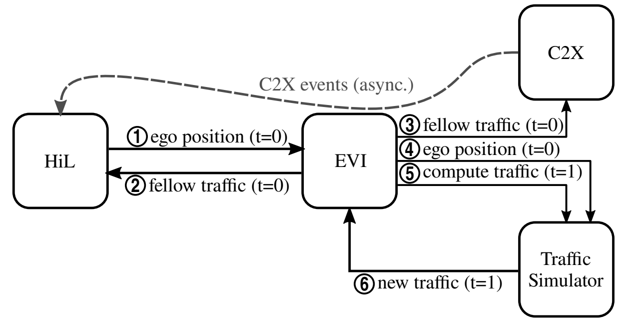


Figure 2. Architecture and data exchange of EVI and coupled simulators, from [4].

EVI[1] acts as a coordinator between the coupled simulators and exchanges updates between them, as shown in Figure 2. Time is split up into synchronization intervals of 100 ms. All coupled simulators perform their simulation for one such time step and then report results back to EVI. The real-time simulator continuously runs and its messages (containing updates in the ego vehicle) serve as real-time triggers. SUMO and Veins wait for such a trigger and updates on the ego vehicle (and traffic for Veins) from EVI after completing their simulation steps. In order to achieve this way of exchanging simulations step results and control time progress, the original connection between OMNeT++ and SUMO in Veins is removed. Instead, all communication between the two passes trough EVI.

EVI is able to filter out a subset of vehicles to send to Veins and the real-time simulator. This is necessary as these two can only work with a limited number of vehicles while maintaining real-time deadlines. The goal is to provide all the data to the ego vehicle that could be simulated if *no* limits to computation time were imposed by real-time deadlines. The EVI uses the subscription method provided by SUMO to obtain updates about the traffic in each time step. Each new vehicle is subscribed to, in order to receive its new position, speed, and status after simulating a time step in SUMO. EVI then stores the vehicle information to pass them to other simulators once they are ready. Coordinated by EVI, SUMO, Veins, and the real-time simulator can run in parallel. For each time step,

[1] More details about EVI available at: http://www.ccs-labs.org/projects/evi

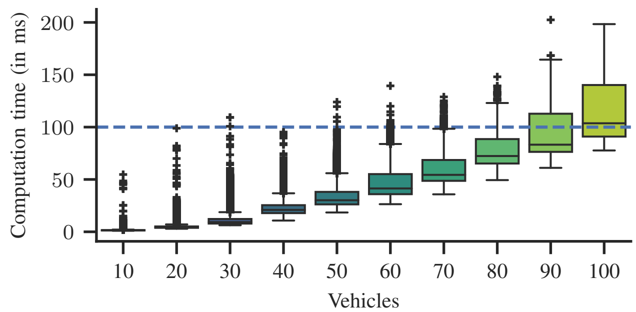


Figure 3. Computation durations of Veins for different amounts of vehicles performing static beaconing at 10 Hz, recorded for time steps of 100 ms. Combined fliers (i.e., outliers) from 8 replications.

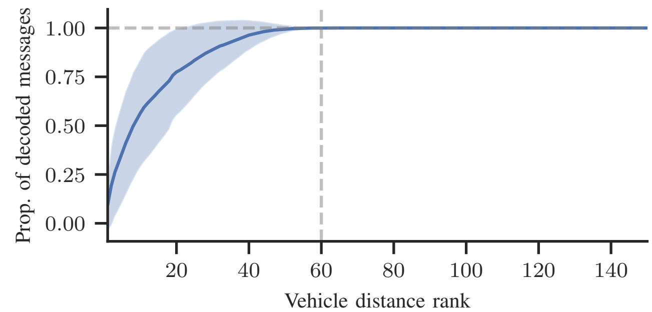


Figure 4. Relation of perceived realism by the ego vehicle to the number of simulated fellow vehicles. Assumes fellow vehicles are sorted by distance to the ego vehicle. The shaded area shows the standard deviation.

EVI already has the traffic update ready from the previous step. This minimizes delay and maximizes the time that can be spent computing the next time step. Once the message denoting the next time step arrives from the real-time simulator, all three simulators are provided with new data.

- The real-time simulator receives traffic updates for its fellow vehicles and results from the V2X application, both stored from the previous step.
- Veins receives updates for the ego vehicle from the message denoting the new time step, and traffic updates computed in the previous step.
- SUMO just receives the update for the ego vehicle.

This update triggers the computation of the next step in SUMO and Veins. Once this is done (for either of them), results are transferred to EVI, converted and stored for later use.

IV. Real-Time Feasibility of V2X Simulation

A. Problem Formulation

V2X simulationulation takes time to compute, which is naturally limited for real-time simulation. This particularly holds for complex and computationally expensive attenuation models such as obstacle shadowing [15]. And with each additionally simulated vehicle, the number of generated messages as well as the number of potential receivers of each message increases. Thus, simulation complexity expected to be $O(n^2)$, with n being the number of simulated vehicles. The exact amount of effort depends on other factors as well, such as the communication protocol and the density of vehicles. At some number of vehicles, the computation time will surpass the deadlines required by real-time simulation. Figure 3 shows an example for Veins, in which an increasing number of vehicles performing static beaconing at 10 Hz (important beaconing protocols, such as *Cooperative Awareness*, have a beacon frequency of up to or below 10 Hz). At about 70 vehicles, the computation time surpasses the simulated time more than just a few times. So, to simulate V2X communication in real-time, the number of simulated vehicles needs to be limited.

The naïve solution of simply limiting the simulated scenario to a small number of vehicles may not work for complex applications or traffic situations. The number of vehicles that could influence the outcome of the simulation may be larger than the number of vehicles that can be simulated in real-time. However, not all vehicle that are simulated may be relevant for the outcome of the simulation. Or vehicles may be only relevant during a portion of the scenario duration, as with EVI, outcome depends on the current location of the ego vehicle.

If the whole scenario could be simulated without real-time constraints, all vehicles would be part of it, naturally including all that are relevant to the outcome of the simulation. Such a simulation would be *complete*, but not real-time feasible. If some vehicles could be excluded from the simulation, or at least for some portion of it, without altering the outcome, the simulation would still be complete, but require less computation duration. Also, some vehicles will contribute only marginally to the outcome of the simulation and thus could be excluded (for some portion) from the simulation while keeping it approximately complete. This way, a reduced simulation of only a subset of vehicles (that changes over time) could be performed in real-time with approximately the same outcome as the original real-time simulation.

To decide if this is actually possible for a given scenario and simulation system (consisting of the simulation software and the used hardware), two core questions have to be answered:

- Which and how many vehicles *need* to be simulated to achieve *completeness*?
- How many messages and vehicles *can* the simulation system simulate in real-time?

B. How to Determine the Number of Vehicles to Simulate

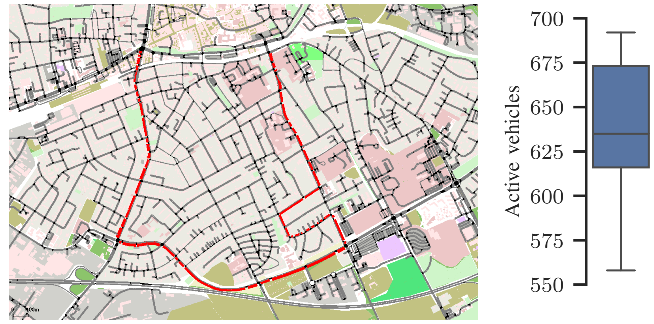
With EVI, the simulation and its outcome are focused on the ego vehicle. If the goal of the simulation is to analyze the behavior of the ego vehicle, only vehicles that can influence the behavior of the ego vehicle need to be simulated. In the V2X simulation, vehicles can only influence the behavior of the ego vehicle, if their messages can be received by the ego vehicle. The chance of any vehicle to communicate with the ego vehicle (i.e., send messages that the ego vehicle can receive) primarily depends on the distance to the ego vehicle. A vehicle close to the ego vehicle will, on average, have a better chance to communicate with the ego vehicle than one far away. Thus, the set of vehicles simulated for V2X simulation should consist of the n vehicles with the lowest distance to the ego vehicle. As the ego vehicle and other vehicles move through the scenario during the simulation, distances to and chances to communicate with the ego vehicle change over time. Thus, the set of vehicles for V2X simulation needs to be adapted at runtime. This metric to select vehicles can be described as a floating ROI around the ego vehicle as shown in Figure 1. However, in an urban scenario with signal shadowing by buildings, closeness and communication range can not be easily quantified as a constant distance. Also, simply selecting all vehicles within a given distance would not limit the number of vehicles, which is necessary for real-time feasibility. So the ROI does not have a fixed diameter, but changes its diameter depending on the position of close-by vehicles.

If too few vehicles are simulated, the ego vehicle will not receive all the messages it would receive in a non-real-time simulation. As the number of simulated vehicles increases, this number of missing messages will decrease. However, at some point, added simulated vehicles will be too far away or behind obstacles, that the number of messages received by the ego vehicle no longer increases. So, at some number of vehicles, the number of messages received by the ego vehicles will converge to the same number that are received in a non-real-time simulation, as shown in Figure 4. To determine the number of simulated vehicles, at which this convergence happens, we use the following procedure:

First, a reference simulation is performed to record which vehicles contribute to simulation completeness (i.e., potentially influence the behavior of the ego vehicle). This reference simulation needs to consider transmissions from all vehicles in the scenario and will not run in real-time. All other simulation parameters that can influence receptions of the ego vehicle need to be the same as in the designated real-time simulations. The ego vehicle can be reduced to a trace or mock up of its mobility is necessary, but it should encounter all traffic situations that will also be relevant in the real-time simulation. From the reference simulation, all transmissions that were successfully received by the ego vehicle need to be recorded. Regardless

Algorithm 1 Calculate number of vehicles for V2X simulation.

Input: successful transmissions to ego vehicle S
Input: acceptable deviation from completeness ϵ
Input: number of aggregation intervals N
  for $t = 0$ to N **do**
    sort vehicles by distance to ego vehicle
    $s \leftarrow \text{count}(S_t)$
    for $v = 0$ to count(vehicles in interval t) **do**
      $T_{t,v} \leftarrow \text{count}(S_{t,v})$
      $C_{t,v} \leftarrow \frac{1}{s} \sum_{k=0}^{v} T_{t,k}$
    end for
  end for
  for $v = 0$ to count(vehicles in simulation) **do**
    $P_v \leftarrow \frac{1}{N} \sum_{k=0} C_{k,v}$
  end for
  return minimal n so that $P_n = 1 - \epsilon$



(a) Highlighted ego vehicle route (b) Active vehicles

Figure 5. Traffic scenario

of transmission success, the distances between each vehicle and the ego vehicle have to be recorded as well. For statistical confidence, the reference simulation should be replicated with different Random Number Generator (RNG) seeds.

From the recordings of the reference runs, the number of vehicles that have to be simulated for completeness can be derived using Algorithm 1. If the proportion of received messages P is plotted over the number of vehicles v, this produces a curve as shown in Figure 4. From this curve, the distance rank or number of vehicles to be simulated can be selected for a desired level of completeness (e.g., 99.9 %). Note that the number produced by this algorithm depends on the scenario. Due to differences in traffic density, road and building layouts, transmit power and receiver sensitivity, etc, it has to be re-computed if the scenario changes.

V. Evaluation

To demonstrate the methods described in this paper and to test the real-time feasibility of Veins, we ran a simulation study for static beaconing in an urban scenario (see Table I). The simulation was performed with EVI coupling SUMO for traffic simulation, Veins for V2X simulation, and a trace from dSPACE ASM™ for the mobility of the ego vehicle.

A. Scenario

For the traffic scenario, we chose a section of the city of Paderborn, a typical mid-sized city in Europe. The scenario represents a normal morning of a weekday without special traffic events. The mobility of the ego vehicle was simulated in dSPACE ASM™ and recorded into a trace of 420 s for deterministic replay. The ego vehicle takes a tour through the south of the city, as shown in Figure 5a. Starting on the junction in the north west, it starts driving south and continues counter-clockwise, while passing different kinds of roads, junctions, and traffic situations. It encounters speed limits of 50 km/h on the first vertical segment, 70 km/h on the southern arterial road, 30 km/h during a detour through the residential area in south east, and back up to 50 km/h for the rest. In total, there are 1676 individual vehicles that are present at some point during the simulation. The number of vehicles present at the same time changes between roughly 550 and 700 (see Figure 5b). Before the ego vehicle starts, the traffic scenario is simulated for 300 s

Table I
SIMULATION PARAMETERS.

Scenario	
Simulation models	SUMO 1.1.0 and Veins 5.0a2
Synchronization interval	0.1 s
Total number of vehicles	1675
Number of present vehicles	558 to 692 (see Figure 5b)
Number of buildings	18784
Traffic warm-up period	300 s
Ego vehicle route duration	420 s
Wireless network simulation	
Simulation models	Veins 5.0a2
Technology	IEEE 802.11p
Carrier Frequency	5.890 GHz
Transmission power	20 mW
Bit rate	6 Mbit/s
Noise floor	−95 dBm
Path loss (Friis model)	$\alpha = 2$
Shadowing (Building loss [15])	$\beta = 9\,\mathrm{dB}$, $\gamma = 0.4\,\mathrm{dB/m}$
Beacon frequency	1 Hz, 2 Hz, 5 Hz, 10 Hz
EVI and simulation control	
Vehicles synchronized to Veins	10 to 100 in steps of 10
Replications	8

to warm up the traffic environment and stabilize the number of vehicles in the scenario. A total number of 18784 buildings are registered as obstacles for wireless communication.

Each simulation run is started by booting EVI, which starts SUMO and Veins and exchanges initial data. The ego vehicle is simulated by a mock-up application, that reads a trace file with recorded behavior from an actual HiL-Simulator (dSPACE ASM™). After each synchronization interval of 100 ms, a message with the current state of the ego vehicle is sent to EVI. The last message of the trace (after 420 s or 4200 messages) contains the de-registration of the ego vehicle, after which EVI shuts down SUMO, Veins, and itself.

The n vehicles with the shortest distance to the ego vehicle (including the ego vehicle) are synchronized to Veins for V2X simulation. We ran the simulation with n between 10 and 100 vehicles. In addition, we performed reference runs in which all vehicles are synchronized to Veins. The results of this reference run are used to validated the completeness of runs with limited vehicles. Each vehicle synchronized to Veins performs static beaconing with a given beacon frequency between 1 Hz and 10 Hz, configured per simulation run. When a vehicle is first synchronized to Veins, it picks a random time between 0 and the beacon interval (the inverse of the beacon frequency). After this random time has passed, it generates its first beacon message and continues to send beacons with exactly one beacon interval in between. This simple beaconing protocol was selected as a stand-in for various V2X applications, such as *Cooperative Awareness*, without introducing artificial behavior of that specific protocol. By assessing different beacon frequencies, the effects on the channel and computation effort can be shown.

All simulations were performed on PCs with an Intel i7 7700K processor and 16 GByte of RAM, running Ubuntu 18.04 (64 bit edition) and using Python 3.6.5. Only one simulation was running at a time per machine, so that no other processes would interfere with the measurement of real-time feasibility. The most relevant simulation parameters are summarized in Table I.

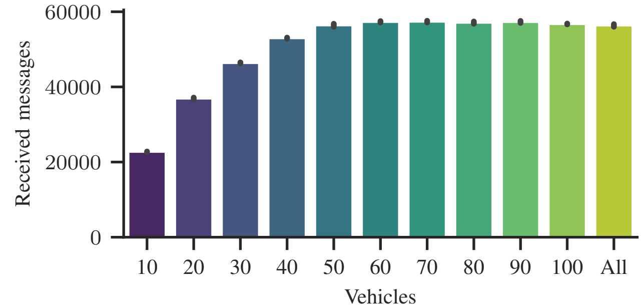


Figure 6. Number of messages received by the ego vehicle for number of simulated vehicles at 10 Hz beacon frequency.

Veins records information about each beacon to determine simulation completeness. For every sent beacon, its unique id, and the id of the sender are recorded, regardless of which vehicle is the sender. The ego vehicle further records information for each beacon that could be potentially received by it: for each potentially received beacon, the unique id, transmission distance, received signal strength, Signal to Interference and Noise Ratio (SINR), and final decision about decodability are recorded. In post processing, these pieces of information are combined to aggregate all information about each beacon that could be received by the ego vehicle. In addition, the beacons are assigned to the unique vehicle id from the traffic simulation. This is necessary, as vehicles in Veins can be assigned different ids if they enter, leave, and re-enter the set of vehicles synchronized to Veins due to their proximity to the ego vehicle. Furthermore, the ego vehicle records statistics aggregated over the whole simulation: the total number of collisions and the channel busy ratio.

The EVI records information about the timing of the coupled simulators as well as its own overhead. For SUMO and Veins, the time it takes them to simulate a synchronization interval (including communication overhead) is recorded.

B. How many vehicles need to be simulated?

To answer the question how many vehicles need to be simulated to approximate simulation completeness in a real-time simulation, we performed the validation method described in Section IV-B. For a first simple check, we sum up the number of beacons successfully decoded by the ego vehicle for each simulation run. We then compare this number over the number of vehicles synchronized to Veins. As shown in Figure 6, this number of successfully decoded beacons converges at around 60 vehicles. A higher number of vehicles synchronized to Veins does not increase the number of decoded beacons, even if all vehicles are simulated. The same holds true for other beacon frequencies (not shown in plots).

To check which vehicles actually communicated with the ego vehicle and whether those were the closest, we analyzed the transmission recordings from the reference simulation runs

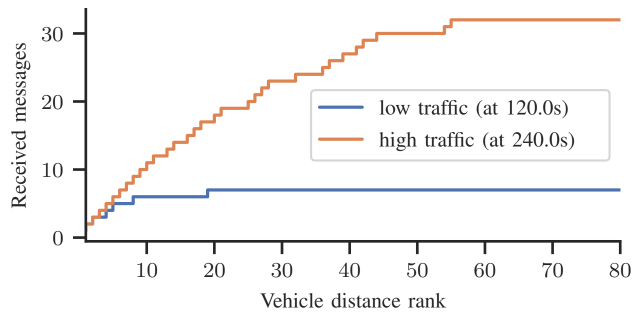


Figure 7. Number of beacons successfully received by the ego vehicle at two specific periods of 100 ms. Once for few (7) successful transmissions at 120.0 s and once for many (32) successful transmissions at 240.0 s (for 10 Hz beacon frequency).

Table II
NUMBER OF VEHICLES NEEDED FOR PROPORTION OF MESSAGES.

Messages	50%	75%	90%	99%	99.5%	99.9%	99.99%
Vehicles	31	42	50	58	60	62	63

with all vehicles synced to Veins. The result is an eCDF of received messages over the distance rank for one period, as shown for two example periods in Figure 7. From this eCDF, we can already see that for the time step at 120.0 s and 240.0 s, 7 and 32 vehicles would have to be simulated, respectively.

To arrive at Figure 4, we average the eCDF over all periods to compute the mean and the standard deviation. Averaging over multiple simulation replications is valid as they all have the same number of periods. From the resulting curve, we can see that, again, at around 60 vehicles, the mean and the standard deviation converge. Vehicles beyond these first 60 only very rarely transmit successfully to the ego vehicle. In other words, simulating more vehicles does not significantly increases simulation completeness.

To quantify the approximation to simulation completeness, we derive the quantiles of successful transmissions over distance rank. As shown in Table II, to simulate 99 % of all messages received by the ego vehicle, 58 vehicles have to be synchronized to Veins. For 60 vehicles, even 99.5 % can achieved.

Another way to verify the necessary number of vehicles to be simulated in Veins is to look at the number of successful communication partners over time. Figure 8 shows how many vehicles successfully sent at least one beacon to the ego vehicle

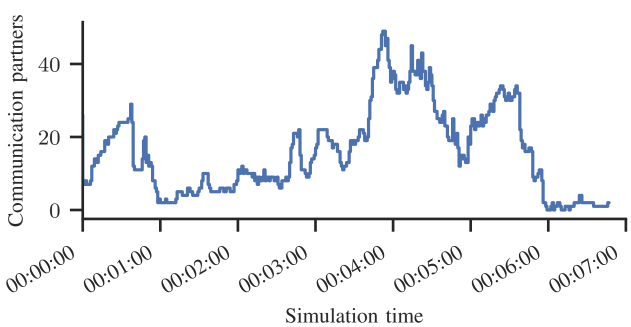


Figure 8. Number of vehicles that successfully transmitted a beacon to the ego vehicle, in intervals of 1 s (for 10 Hz beacon frequency).

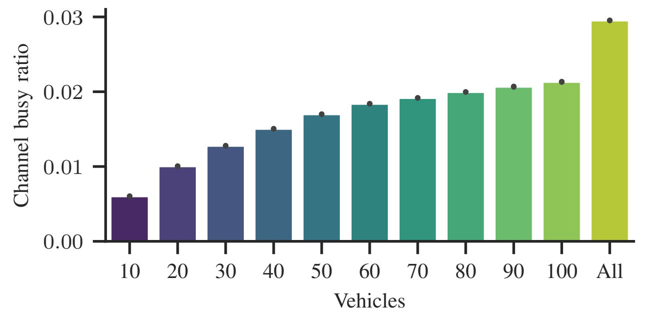


Figure 9. Channel busy ration as perceived by the ego vehicle (for 10 Hz beacon frequency).

in a period of 1 s in one reference run. As the plot shows, the number of successful communication partners stays below 50. If it was higher than the number of vehicles to be simulated in Veins, this would indicate a problem, unless the aggregation period for the plot would be much larger that the beacon frequency and/or the synchronization interval.

While the number of messages received by the ego vehicles can be approximated even with around 60 vehicles, the channel busy ratio perceived by the ego vehicle, does not converge like that. As shown in Figure 9, the channel busy ratio continues to increase with an increasing number of vehicles synchronized to Veins. Also, the channel busy ratio recorded in the reference simulation is almost twice as high as the value for 60 vehicles and around 150% of the value for 100 vehicles (the highest parameter tried). This also explains why the total number of messages received by the ego vehicle shown in Figure 6 slightly decreases after it reached its maximum at around 60 vehicles. Some messages that were successfully received in the real-time simulation with, e.g., 60 vehicles, were not successfully received in the reference simulation due to collisions and busyness on the wireless channel. However, we consider this difference small enough to be acceptable.

C. How many vehicles can be simulated?

To answer the question how many vehicles can be simulated in real-time, we measured the answer time of Veins for each time step for different numbers vehicles simulated. As Figure 3 shows for an example of 10 Hz beacon frequency, the computation times increases steadily with the number of vehicles simulated. The increase indeed implicates a computation complexity of $O(n^2)$. This matches our previous assumption that each vehicle increases both the number of beacons sent and the number of possible receivers (i.e., the number of one-to-one transmission per sent beacon).

As shown in Figure 10, between 60 and 70 vehicles can be simulated without breaking the deadline of 100 ms regularly for a beacon frequency of 10 Hz. For a beacon frequency of 5 Hz, the number of vehicles increases to 80 to 90, while for even lower beacon frequencies, the number of vehicles is above 100. However, this only shows the computation duration of Veins and does not incorporate all overhead of EVI, as only the answer time of Veins to EVI is shown in the plot.

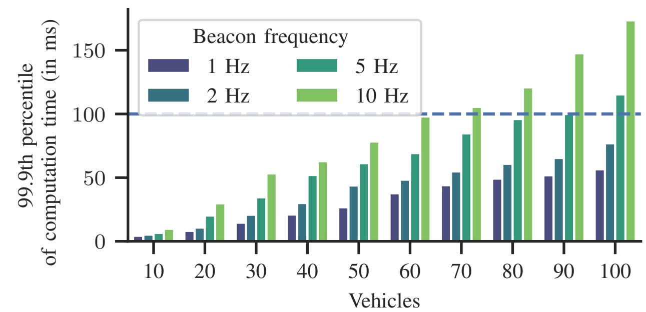


Figure 10. 99.9th percentiles of the computation times of Veins for different beacon frequencies and numbers of vehicles.

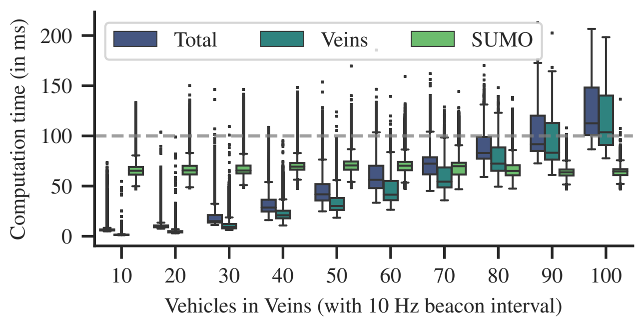


Figure 11. Computation durations or answer times of EVI (*Total*, includes internal processing and Veins), Veins, and SUMO. Note that SUMO is computed one step in advance, so the duration does not add up to the *Total* duration.

While most of the evaluation focused on the computation duration of V2X communication in Veins, EVI, SUMO, and the data exchange overhead still have to be considered for real-time feasibility. As described in Section III, EVI exchanges simulation results between the simulators each time step. After the real-time simulator signaled a new time step, Veins performs the simulation for that time step in parallel to the real-time simulator, while SUMO pre-computes the traffic data for the next time step. To achieve real-time feasibility, all these steps have to finish in time, with enough buffer left for EVI to collect the results and prepare them for the real-time simulator. We assume the real-time simulator is able to send its update messages of the ego vehicle (which trigger the next time step) at a regular interval.

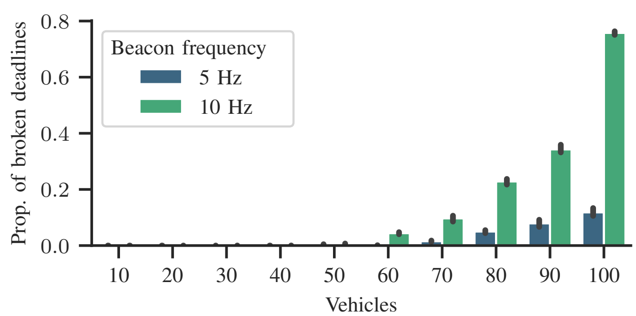


Figure 12. Proportion of time steps with broken deadlines (i.e., more than 100 ms answer time) of EVI. Error bars over repetitions. Beacon frequencies of 2 Hz and 1 Hz not shown as all deadlines were kept.

As Figure 12 shows, EVI can keep its real-time deadlines until 50 to 60 vehicles in Veins for 10 Hz beacon frequency. Up until 80 vehicles, these broken deadlines can be recovered in the following steps (not shown in plots). But at 90 vehicles, the proportion of deadlines missed and the amount of time they are missed (see the distribution at 90 vehicles in Figure 11) breaks even soft real-time. For lower frequencies, these limits shift towards more vehicles without losing real-time feasibility.

SUMO can simulate thousands of vehicles at 100 ms update rate in real-time. There are some spikes in the computation duration due SUMO loading the next batch of input data, such as vehicles and routes. However, the overhead of processing results sent by SUMO is much more of a problem for EVI. Simply receiving and converting updated vehicle data (using the official Python TraCI library) takes a long time once there are more than 100 vehicles. This limited the size of the total scenario compared to what SUMO itself could actually support in real-time. But, as Figure 11 shows, the computation duration of SUMO (including result processing in EVI) does not change if the number of vehicles in Veins increases.

VI. Discussion

The presented results have shown that the necessary number of vehicles to be simulated for this scenario is around 60 vehicles and that Veins and EVI can simulate this amount even for 10 Hz beacon frequency. For more than 99.9 % of the messages, the real-time deadline was not broken by Veins and EVI could answer quickly to the real-time simulator. But the simulators Veins and SUMO, as well as the frameworks they depend on, were not built to be real-time compliant. So there can be time steps in which additional computations occur, which increase the computation duration beyond the real-time deadline. Examples for this could be I/O operations for result collection or interference from the operating system. Yet, as long as most of the simulation steps stay below the real-time deadline, overstepping the deadline in one step can be recovered in the next step [5]. This prevents chain-reactions in which the time by which the deadline is overstepped increases with every simulation step and the whole simulation becomes invalid. However, some applications may require that real-time deadlines are never missed. In such cases, the simulation will have to be repeated, potentially with a reduced scenario. But for many cases in which only soft real-time feasibility is required, such as when coupling to a driving simulator, a temporary overstepping can be tolerated.

The traffic scenario selected for the experiment aims to resemble a situation in a medium-sized city in Europe. During the simulation, the total number of vehicles in the whole traffic scenario says relatively stable. However, the number of vehicles that actually communicate with the ego vehicle changes over its course through the scenario (see Figure 8). Figure 7 shows how the number of vehicles that successfully communicate with the ego vehicle and thus are necessary for simulation completeness may vary over different situations in the scenario. The same figure also shows that the vehicles communicating successfully with the ego vehicles are not just the absolutely closest ones.

There are horizontal stretches in the curves, showing that there are some vehicles are not successfully transmitting to the ego vehicle despite being closer than many others. This suggests that there may be more effective ways to determine which vehicles are relevant to the ego vehicle compared to picking the n closest ones by Euclidean distance.

The communication pattern selected for the experiment, static beaconing, can be used to gain estimated insights in a various beaconing protocols. Still, even in the reference simulation the mean channel busy ratio measured by the ego vehicle is only around 3 %. In scenarios with higher channel utilization, effects like interference, collisions, and time spent waiting to access the channel may become more dominant and limit the validity of the successful transmissions as a metric for simulation completeness. The effects on communication patterns with reactive behavior remain as future work.

The results also show that EVI and Veins are feasible to perform real-time experiments in urban scenarios. But these would not have been possible without the recent improvements of the wireless signal representation [16] in Veins 5.0a1 and the obstacle shadowing model in Veins 5.0a2. Yet there are still issues that need to be approached for more stable and less limited real-time simulations. The large overhead of the TraCI protocol implementation in EVI prevents larger or denser traffic scenarios. The throughput of Veins is still limited mostly by the attenuation models. Especially the obstacle shadowing model consumes the major part of the computation time but could benefit significantly from parallelization. There might as well be other attenuation models relevant for different scenarios, such as vehicle shadowing (e.g., for platooning). Or even different communication technologies than IEEE 802.11p, such as cellular technologies or vehicular visible light communication. Solving these issues could open up the technique of real-time interactive V2X simulation to more complex scenarios and applications.

VII. Conclusion

In this paper, we presented a method to run real-time interactive V2X simulation for an ego vehicle. The ego vehicle could be controlled using HiL solutions or simply a driving simulator enabling a user to interact with the vehicular network simulation in real-time. We derived a concept to validate the feasibility of such a focused real-time V2X simulation and developed a method to identify the amount of fellow vehicles that need to be simulated for simulation completeness. Our results show that EVI can perform a real-time feasible simulation of an urban scenario. Around 60 vehicles were necessary to perform a simulation that, to the ego vehicle, was sufficiently close to a full, non-real-time simulation. Veins proved to be able to simulate these 60 vehicles running a static beaconing protocol with up to 10 Hz in real-time.

The concept to validate the feasibility of focused real-time V2X simulation was designed for urban scenarios with one-hop protocols. While it is probably applicable to highway scenarios as well, more complex protocols such as platooning, geocasting, or maneuver coordination can introduce dependencies that have to be examined more closely in future work. In future work, we want to assess if different vehicle selection mechanisms can reduce the number of vehicles or otherwise improve simulation completeness. Using parallel and distributed simulation of V2X, larger and denser scenarios as well as more complex scenarios could be achieved. This could prove especially useful for scenarios involving multiple ego vehicles.

Acknowledgements

Research reported in this paper was conducted in part in the context of the Hy-Nets4all project, supported by the European Regional Development Fund (EFRE-0801678).

References

[1] Z. MacHardy, A. Khan, K. Obana, and S. Iwashina, “V2X Access Technologies: Regulation, Research, and Remaining Challenges,” *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, 2018.

[2] J. Guanetti, Y. Kim, and F. Borrelli, “Control of connected and automated vehicles: State of the art and future challenges,” *Annual Reviews in Control*, vol. 45, 2018.

[3] J. Wang, Y. Shao, Y. Ge, and R. Yu, “A Survey of Vehicle to Everything (V2X) Testing,” *MDPI Sensors*, vol. 19, no. 2, 2019.

[4] D. S. Buse, M. Schettler, N. Kothe, P. Reinold, C. Sommer, and F. Dressler, “Bridging Worlds: Integrating Hardware-in-the-Loop Testing with Large-Scale VANET Simulation,” in *IEEE/IFIP WONS 2018*, Isola 2000, France: IEEE, Feb. 2018.

[5] S. Henning, D. S. Buse, M. Franke, A. Trächtler, S. Gausemeier, and F. Dressler, “Proof-of-Concept einer komplexen Co-Simulationsumgebung für einen Fahrsimulator zur Untersuchung von Car2X-Kommunikations-Szenarien,” in *AUTOREG 2019*, Mannheim, Germany: VDI, Jul. 2019.

[6] C. Sommer, 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, Jan. 2011.

[7] C. Obermaier, R. Riebl, and C. Facchi, “Fully Reactive Hardware-in-the-Loop Simulation for VANET Devices,” in *IEEE ITSC 2018*, Maui, HI: IEEE, Nov. 2018.

[8] S. Laux, G. S. Pannu, S. Schneider, J. Tiemann, F. Klingler, C. Sommer, and F. Dressler, “OpenC2X - An Open Source Experimental and Prototyping Platform Supporting ETSI ITS-G5,” in *IEEE VNC 2016, Demo Session*, Columbus, OH: IEEE, Dec. 2016.

[9] Z. Szendrei, N. Varga, and L. Bokor, “A SUMO-Based Hardware-in-the-Loop V2X Simulation Framework for Testing and Rapid Prototyping of Cooperative Vehicular Applications,” in *VAE 2018*, K. Jármai and B. Bolló, Eds., Miskolc, Hungary: Springer, Cham, May 2017.

[10] P. Alvarez Lopez, M. Behrisch, L. Bieker-Walz, J. Erdman, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, “Microscopic Traffic Simulation using SUMO,” in *IEEE ITSC 2018*, Maui, HI: IEEE, Nov. 2018.

[11] M. Aramrattana, T. Larsson, J. Jansson, and A. Nåbo, “A simulation framework for cooperative intelligent transport systems testing and evaluation,” *Transportation Research Part F: Traffic Psychology and Behaviour*, 2017.

[12] E. Egea-Lopez, F. Losilla, J. Pascual-Garcia, and J. M. Molina-Garcia-Pardo, “Vehicular Networks Simulation With Realistic Physics,” *IEEE Access*, vol. 7, Apr. 2019.

[13] Y. Zhao, A. Wagh, Y. Hou, K. Hulme, C. Qiao, and A. W. Sadek, “Integrated Traffic-Driving-Networking Simulator for the Design of Connected Vehicle Applications: Eco-Signal Case Study,” *Journal of Intelligent Transportation Systems*, 2016.

[14] B. Sliwa, J. Pillmann, F. Eckermann, L. Habel, M. Schreckenberg, and C. Wietfeld, “Lightweight joint simulation of vehicular mobility and communication with LIMoSim,” in *IEEE VNC 2017*, Turin, Italy, Nov. 2017.

[15] C. Sommer, D. Eckhoff, R. German, and F. Dressler, “A Computationally Inexpensive Empirical Model of IEEE 802.11p Radio Shadowing in Urban Environments,” in *IEEE/IFIP WONS 2011*, Bardonecchia, Italy: IEEE, Jan. 2011.

[16] F. Bronner and C. Sommer, “Efficient Multi-Channel Simulation of Wireless Communications,” in *IEEE VNC 2018*, Taipei, Taiwan: IEEE, Dec. 2018.