Coupled Simulation of Mobile Cellular Networks, Road Traffic and V2X applications using Traces

Norbert Goebel Raphael Bialon Martin Mauve Kalman Graffi
Department of Computer Science, University of Düsseldorf, Germany
{goebel, bialon, mauve, graffi}@cs.uni-duesseldorf.de

***Abstract*—The development and evaluation of *Vehicle-to-X* (V2X) applications using mobile cellular networks by means of field tests is time consuming and expensive. Simulations can speed up the development and largely reduce evaluation costs. However, due to the complex nature of the network involved, it is quite difficult to be certain that the results obtained using simulation will actually match the observable behaviour in the real world. In this paper, we introduce a novel trace-based simulation environment for V2X applications using mobile cellular networks. It employs real world measurement traces as a basis and thus avoids many uncertainties of other simulation approaches.**

I. Introduction

Estimating the impact of a particular V2X application is a pivotal question in the context of vehicular communication. While this question can be addressed using field tests like *Sichere Intelligente Mobilität Testfeld Deutschland* (sim$^{\text{TD}}$) [1] where a large number of equipped vehicles conduct planned experiments in various locations, those field tests are costly and time consuming. It is quite unlikely that this approach can be used for each V2X application that is developed in the future.

As an alternative to field tests during V2X application development, simulations could be used. However, the maturity and accuracy of simulation approaches for V2X applications is very heterogeneous, depending on the underlying communication technology. While many proven and tested simulation models and implementations for 802.11p wireless LAN communications exist, the situation is quite different with regards to mobile cellular networks. The main reasons for this are the very complex nature of cellular wireless networks, which makes them hard to model and simulate accurately, and the requirement for in-depth information of the cellular networks, which the ISPs keep as company secrets.

In this paper, we introduce a trace-based simulation environment for V2X applications that is capable of simulating today's cellular network communication using real world measurements. The key contributions of this paper are:

- The introduction of a complete simulation infrastructure allowing coupled simulation of realistic mobile cellular network communication, road traffic and V2X applications. Our solution allows the cost effective implementation and repeatable evaluation of V2X applications using cellular network communication by utilizing network characteristics observed in the real world.
- Furthermore, the *Objective Modular Network Testbed in C++* (OMNeT++) implementation of our trace-based simulation model for cellular networks can be used in any simulation relying OMNeT++ as a network simulator.
- The implementation of a V2X application defined by the European Telecommunications Standards Institute showing the usability of our simulation tool chain.
- The comparison of the simulation results of the *Emergency Warning Application* (EWA) and a modified, cellular network optimized, version of the EWA. With this example use case, we show the need for good simulations for the development of new V2X applications using cellular networks. Furthermore we outline how valuable simulations are when porting existing V2X applications, specified for 802.11p networking, for cellular networks.

We start the remainder of this paper by presenting the complete simulation architecture in Section II. Further on we explain how the simulation database is generated in Section III. In Section IV we explain the basic aspects of our simulation model. In Section V we discuss changes to the simulation framework and OMNeT++ which were needed to allow graph-based position distribution. Finally, we specify the example V2X application used to demonstrate the functionality of our complete simulation tool chain in Section VI. The simulation results are presented and evaluated in Section VII. Section VIII presents the related work. We conclude the paper with Section IX.

II. Simulation Architecture

The basic idea of our simulation approach is to use trace-based simulation of a mobile cellular network and to combine it with existing simulators to gain a complete simulation infrastructure yielding realistic simulation results.

Figure 1 depicts our simulation environment. All components implemented or enhanced by us, as described in the paper at hand, are marked in red. We use the *V2X Simulation Runtime Infrastructure* (VSimRTI) [2] to couple all involved simulators. *Simulation of Urban Mobility* (SUMO) [3] is used for traffic simulations. We make use of the *V2X application simulator* (VSimRTI_App) for V2X application simulation. The mobile cellular network characteristics are simulated by OMNeT++ [4], which uses our OMNeT++ module, *libtbus*.

In the following four sections, we explain — from bottom to top — all steps necessary to add trace-based simulation to the simulation framework. We start by explaining how we conduct the network measurements and how we map-match and aggregate the results in our graph-based database.

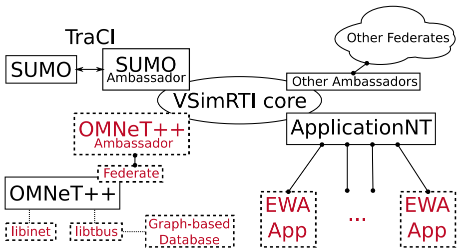


Figure 1. Structure of our simulation environment.

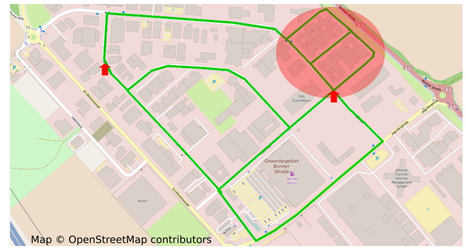


Figure 2. Excerpt showing the simulation area (Map © OpenStreetMap [6]).

Next, we explain our trace-based simulation model, which we implemented as the OMNeT++ module *libtbus*. Further on we briefly describe changes needed regarding the *V2X Simulation Runtime Infrastructure* (VSimRTI) framework itself and finally introduce an exemplary V2X application use case. This use case is then examined using our simulation approach.

III. Generating the Simulation Database

Our simulation approach relies on accurate network characteristics for every road segment of the road network graph in the simulation area. We utilize SUMO's underlying structure which is built upon a directed graph.

Figure 2 shows an exemplary simulation area, where network characteristics were gathered for every allowed driving direction of the roads marked in green, which later on were used in the simulation. The network characteristics used by the simulation model are the available data rate, the delay and the loss probability. Furthermore the currently used cell sector of the mobile cellular network needs to be identified and logged, as it is used in the simulation model, too. Since all gathered data has to be mapped onto the road network graph, it is further necessary to trace the position of the measurement vehicle.

In the next section, we explain how we conduct the network measurements. This is followed by a section describing how we map the data to the road network graph and how we aggregate multiple measurements of the same road segment.

A. Measurements

To achieve realistic cellular network simulation, a measurement method, allowing fast consecutive measurements of the delay, the available data rate and the loss probability for both communication directions is needed. Additionally, the network cell sector currently used by the cellular modem has to be stored. Furthermore, the vehicle's position has to be logged to allow to map the measurement data to the road network graph.

We examined numerous papers in the field of network measurements and while the measurement of the one-way delay and loss probability has been sufficiently covered, the measurement of the available data rate in cellular networks is quite challenging. All relevant existing methods can be divided into three groups using either the *Probe Gap Model (PGM)*, the *Probe Rate Model (PRM)* or *Bulk Traffic*. But none of the existing solutions is capable of simultaneous high frequency measurements of delay, available data rate and loss probability on high volatile networks like cellular networks. Hence, we introduced and evaluated the *Rate Measurement Framework* (RMF) in a previous paper [5]. The RMF is capable of high frequency simultaneous measurements of the available data rate, the loss rate and the delay for each communication direction of the network path between a mobile node, using mobile cellular networks, and an Internet homed server. It is able to sustain a measurement frequency of 4 to $5\,\mathrm{Hz}$ most of the time except in areas of no reception — or areas where large packet delays are abruptly inserted into the measurement traffic by the cellular network.

While measuring the network characteristics we simultaneously collect the *Global Positioning System* (GPS) positions of the measurement vehicle with a frequency of $1\,\mathrm{Hz}$ and the currently used cell sector with a frequency of 5 to $10\,\mathrm{Hz}$. All collected data is logged with a timestamp in nanoseconds.

B. Map-Matching

The GPS noise and the diverging time stamps of the network characteristics, the GPS positions and the used cell sectors necessitate the post-processing of the data. The graph-based simulation database our simulation model uses, is built upon the following tuples (e, o, g, t, a, d, l, c). Each tuple contains the **e**dge ID, the **o**ffset on the edge, the **g**roup ID, the measurement **t**ime, the **a**vailable data rate, the backbone **d**elay, the **l**oss probability and an ID representing the used **c**ell sector. An edge ID identifies a road segment and the driving direction.

To generate these tuples, multiple post-processing steps are needed. In the first step we map-match every collected GPS position onto a road network graph derived from OpenStreetMap's [6] OSM files using a modified version of the N-Route Algorithm derived from [7]. We thereby gain an edge sequence of the street graph, that represents the route traveled by the measurement vehicle. The switch from latitude/longitude coordinates to a representation using the edge ID and the offset from the start of this edge adds the movement direction to the information base of the simulation. This enhances simulation accuracy in regard to cell sector changes, which are directly influenced by the driving direction. The gathered modem information on the used network cell — frequency and primary scrambling code in 3G networks and location area code and cell ID in 2G networks — allows the identification of the cell sector used for each network measurement. This leads to tuples (e, o, t, a, d, l, c).

In a second step, we deal with multiple passes of a single road segment during the measurements. It is perfectly possible — and in many cases intended — that a road segment is passed multiple times during a measurement drive, or that multiple measurements are conducted in the same area. In Figure 3 an example situation is sketched, where one edge is passed two

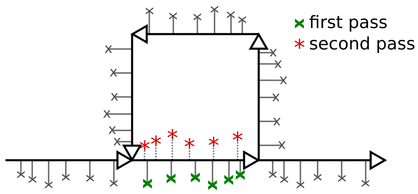


Figure 3. Example of an edge that is passed two times.

times. The measurements of the first pass are represented by an x and those of the second pass are represented by a $\star$. Imagine the first pass of the edge in our example is conducted at midnight and the second pass during rush hour. It is obvious, that the measured network characteristics can deviate largely.

Thus, directly merging network measurements of the same edge from different traces can result in multiple switches between extremely different network characteristics. In the example shown in Figure 3 this leads to the sequence ⋆x⋆x⋆⋆xx⋆x⋆x. This can significantly influence the simulation outcome and lead to unrealistic network simulations, as those measurements were not observed in direct sequence.

Hence network characteristics gathered from measurement drives are separated into *value groups*. With each value group containing network characteristics from one continuous drive on an edge, a unique group id g is added to every measurement of the value group. For every network measurement the post-processing thus results in the desired tuple (e, o, g, t, a, d, l, c).

To ease the selection of specific network characteristics for edges with multiple group IDs, we added an optional third post-processing step. We implemented a filter that allows the selection of a specific group per edge for the simulation. This filter allows the selection of groups by parameters, like for example, “worst measured average delay”, “highest loss rate” or “lowest available data rate”. Thus the combination of multiple traces by choosing network characteristics, that are most interesting for the desired simulation scenario, but still retaining realistic network simulations, is possible.

All generated tuples (e, o, g, t, a, d, l, c) are stored in the graph-based simulation database. The same graph used for the whole map-matching process is also used by the traffic simulator SUMO as its road network graph.

IV. The TBUS Simulation Model

In [8] we introduced the *Trace Based UMTS Simulation* (TBUS) model for the simulation of mobile cellular networks using real world traces. Here, we use this model to simulate the traversal of packets through the network based on the previously generated graph-based database. TBUS uses two main assumptions: (1) the wireless link is the *tight link* and (2) the majority of packet drops occur on the wireless link.

Given these assumptions, we built the simulation model shown in Figure 4. It simulates the upstream (left) and the downstream (right) using two sequentially traversed queues each: the queues (CRSQ/CRRQ) simulate the available data rate (and thus indirectly the transmission delay) and the packet loss, the queues (CDSQ/CDRQ) simulate the backbone delay.

In the TBUS model, every position change of a simulated vehicle triggers an update of the simulated network

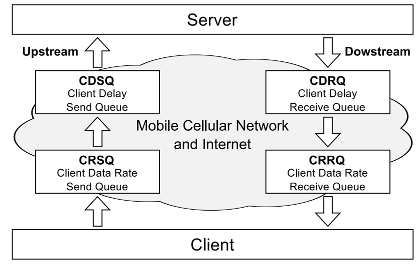


Figure 4. Simulation model for client-server communication.

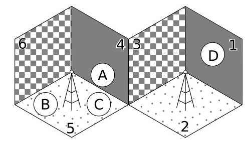


Figure 5. Sketch of nodes in different cell sectors.

characteristics for this vehicle, as long as there are differing values stored for the new position in the database. In the paper at hand we implemented the simulation model as an OMNeT++ module and further enhanced it by incorporating a cell share model introduced in the following subsection. Our implementation assumes that all vehicles use the same cell network. Handling multiple cell networks can be implemented by using multiple databases and linking the queues of each vehicle to the database of the corresponding cell network.

A. Cell Share Model

In cellular networks, a cell is further divided into *cell sectors*. A node is connected to one *cell sector* of a network cell, belonging to a base station. This partitioning is sketched in Figure 5, where node *A* is in cell sector 4 and nodes *B* and *C* are in sector 5 of the same cell. The other cell only hosts node *D*, connected to its sector 1. Sectors 2, 3 and 6 have no connected nodes.

Data is sent to and received from the base station and as nodes belonging to the same cell sector share a particular entity of the cell, these nodes’ sending and receiving behaviour influence each other’s network characteristics. To accommodate this behaviour, we introduce the *cell share model* as an intermediate processing layer between the measured available data rate and the available data rate used in the simulation. Cell sector affiliation is monitored by an omniscient module, which then calculates a node’s share of the measured available data rate according to the current situation in the cell sector.

The cell share model provides an open interface for a thorough calculation of interferences, so that the division of shares can be controlled for each occasion individually. The interface is also set up modularly and can be exchanged by implementations matching the measured network situation, or even take further influence on network behaviour. For the simulations used in the paper at hand, we implemented a fair share model, where, given that $n > 1$ vehicles are actively transferring data in the same cell sector, each vehicle is simulated with an available data rate $A_{\text{simulation}} = \frac{A_{\text{measured}} \cdot 110\%}{n}$. Nothing is changed for $n = 1$. As the simulation uses different available data rate measurements per direction, the calculations of n and $A_{\text{simulation}}$ are communication direction dependent.

V. VSimRTI and OMNeT++ Modifications

Previous simulations within the VSimRTI utilize positions given as points on a Cartesian plane. To allow the TBUS model to operate on graph-based positions, this position type has to be propagated to all simulators used by the VSimRTI framework. Existing components relying on Cartesian positions have to be considered and compatibility to preceding simulations has to be maintained. As SUMO operates on a graph-based road network and VSimRTI core is aware of both position types, too, acquiring graph-based positions is straight forward. Only the connection protocol between VSimRTI core and OMNeT++ had to be extended. A new *mobility* model is introduced in order to maintain the desired compatibility.

As TBUS is implemented as an OMNeT++ library, *libtbus*, which only depends on OMNeT++ and the *INET framework for Mobile Ad-Hoc Networks* (INETMANET) modules, no further parts of VSimRTI have to be touched. This leads to a small patch of the OMNeT++-Ambassador and -Federate.

The aforementioned modifications will be integrated into the next official VSimRTI release in cooperation with Fraunhofer FOKUS. Our current implementations, based on VSimRTI version 0.14.0, are either available through the authors' website[1], or through links on that web page. They are released under the MIT license except for the OMNeT++-Ambassador, which is based on closed source of Fraunhofer FOKUS and thus is only available as a binary through their website. All map-matched traces and simulation scenarios needed to reproduce our simulation results are also available for download.

VI. Example V2X Use Case

In [9], the *European Telecommunications Standards Institute* (ETSI) defines many possible V2X applications. We want to show a use case that is simple to understand — in theory and by reading our source code. We thus scanned the possible use cases and, with use case C1.2.1, the Emergency Warning Application, chose an especially interesting one. It is completely defined for 802.11p, but lacks messaging interval definitions for cellular networks.

In the remainder of this section, we discuss our implementation of a geosever and geoclient, explain the implementation of the ETSI defined *Emergency Warning Application* (EWA) and finally introduce an optimized EWA that is much better suited for cellular network usage. We end this section by a description of the simulation area used for the evaluation.

A. Geoserver and Geoclient

Contrary to ad-hoc scenarios, messages via conventional mobile cellular networks cannot simply be broadcasted. Hence the ETSI defined the application infrastructure to be used in V2X scenarios in [9, section 6.1]: a Geomessaging Enabler and a Geocast Client. We implemented the geoserver to act as the Geomessaging Enabler and the geoclient as the Geocast Client. The geoserver implementation monitors node position updates and coordinates geo-broadcast distribution.

It has to be noted that our implementations' broadcasting technique and routing logic differ slightly from those defined by ETSI. While the ETSI specifies different geometrical shapes as broadcast target areas in [10, section 4] and [11], we chose to provide a novel broadcasting method based on the road network graph. This leads to a geo-broadcast distribution realized using graph-based routing. The geoserver forwards geo-broadcasts to all vehicles that can cross the emergency vehicle's way within $50\,\mathrm{m}$ of the end of the emergency vehicle's current edge. Not only does this fit our graph-based simulation structure, it also reduces the number of messages in the network and thus the overall bandwidth usage.

We created a framework for TBUS-based applications constructed upon the geoserver and geoclient, offering basic functionality and utilizing ETSI defined message types. V2X applications extending this framework inherit an integrated periodical position update mechanism and functionality to send geo-broadcasts so that they can fully focus on their application logic on a higher layer. Messages for position updates are modeled as *Cooperative Awareness Messages* (CAMs) and informative broadcasts are modeled as *Decentralized Environmental Notification Messages* (DENMs) according to ETSI standards defined in [12], [13]. CAMs have a payload size of 200 Byte and DENMs have a payload size of 40 Byte.

B. Emergency Warning Application

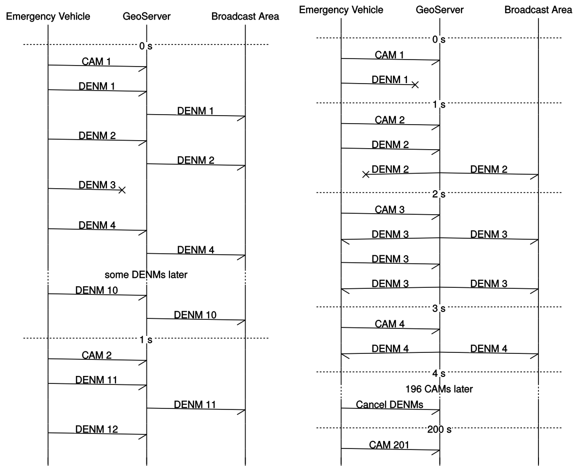
According to ETSI use case C.1.2.1 as described in [9], we introduce the *Emergency Warning Application* (EWA) as an example implementation of a TBUS-based V2X application.

Every vehicle in the simulation announces its position with a frequency of $1\,\mathrm{Hz}$ to the geoserver using CAMs. In addition to the CAMs, emergency vehicles send warning messages to the geoserver using DENMs with a frequency of $10\,\mathrm{Hz}$, as suggested by the ETSI. An example message flow for the messages sent by an emergency vehicle is given in Figure 6a. It shows that the geoserver relays each DENM to all vehicles within the broadcast area — evaluated by using their last known position received via CAM. Each regular vehicle equipped with an instance of the EWA reacts to received geo-broadcasts by slowing down to a halt and letting emergency vehicles pass. All vehicles are simulated with respect to SUMO's default traffic laws and driver models. Thus equipped emergency vehicles do not impede each other and so multiple emergency vehicles can be simulated simultaneously.

C. Enhanced Cellular Network Aware EWA Application

Apparently the high frequency of $10\,\mathrm{Hz}$ for the DENMs were introduced by the ETSI with 802.11p in mind, where interference, shadowing and hidden terminals have great impact on the reception probability of messages. In cellular networks, packet loss is much less severe, except in areas of bad cellular network coverage, where often no communication is possible at all. As the high repetition rate of $10\,\mathrm{Hz}$ is not caused by the need of high frequency position updates of the emergency

[1]https://wwwcn.cs.uni-duesseldorf.de/software/simulation.html



(a) ETSI EWA (b) Enhanced EWA

Figure 6. Emergency vehicle CAM and DENM distribution chart.

vehicle, we implemented a second, cellular network aware version of the EWA.

In the enhanced EWA, we reduce cellular network traffic by shifting intelligence to the geoserver. Figure 6b shows an example message flow of messages created by or on behalf of the emergency vehicle. All vehicles, including emergency vehicles, keep sending $1\,\mathrm{Hz}$ position updates as CAMs to the geoserver. While the geoserver still relays DENMs to their designated broadcast areas, five simple modifications offload the sending of DENMs mostly to the geoserver.

- At the beginning of an emergency, the emergency vehicle sends DENMs in $1\,\mathrm{Hz}$ intervals (DENMs 1 to 3).
- On reception of a DENM, the geoserver relays the message to all vehicles in the broadcast area, including the emergency vehicle. And the geoserver stores the unique emergency ID and marks the emergency vehicle as active.
- On the reception of a regular $1\,\mathrm{Hz}$ CAM of an active emergency vehicle, the geoserver generates a new DENM on behalf of the emergency vehicle and sends it to the updated broadcast area (CAM 4 and DENM 4).
- An emergency vehicle ceases sending DENMs for $5\,\mathrm{s}$ after receiving a DENM containing its emergency ID.
- The server stops generating DENMs on behalf of an emergency vehicle and marks it inactive either after $5\,\mathrm{s}$ without new CAMs from the emergency vehicle or after receiving a special *cancel* DENM.

These changes suffice to largely reduce the bandwidth consumption in the upstream direction in cell sectors with active emergency vehicles and thus reduces the risk of severely overloaded cell sectors in the upstream direction. Furthermore, the reduced DENM frequency of approximately $1\,\mathrm{Hz}$ also reduces the bandwidth consumption in the downstream.

D. Simulation Scenario

As an example simulation area, we chose the industrial area *Taubental* in 41468 Neuss, Germany, depicted in Figure 2.

It offers a large enough area for the simulation of a larger number of vehicles with different cells of the mobile cellular network. The circled area includes most of the area's crossings and the efficacy of the EWA can best be observed in this area. Vehicles drive along routes originating in the position of the right arrow and ending at the position of the left one.

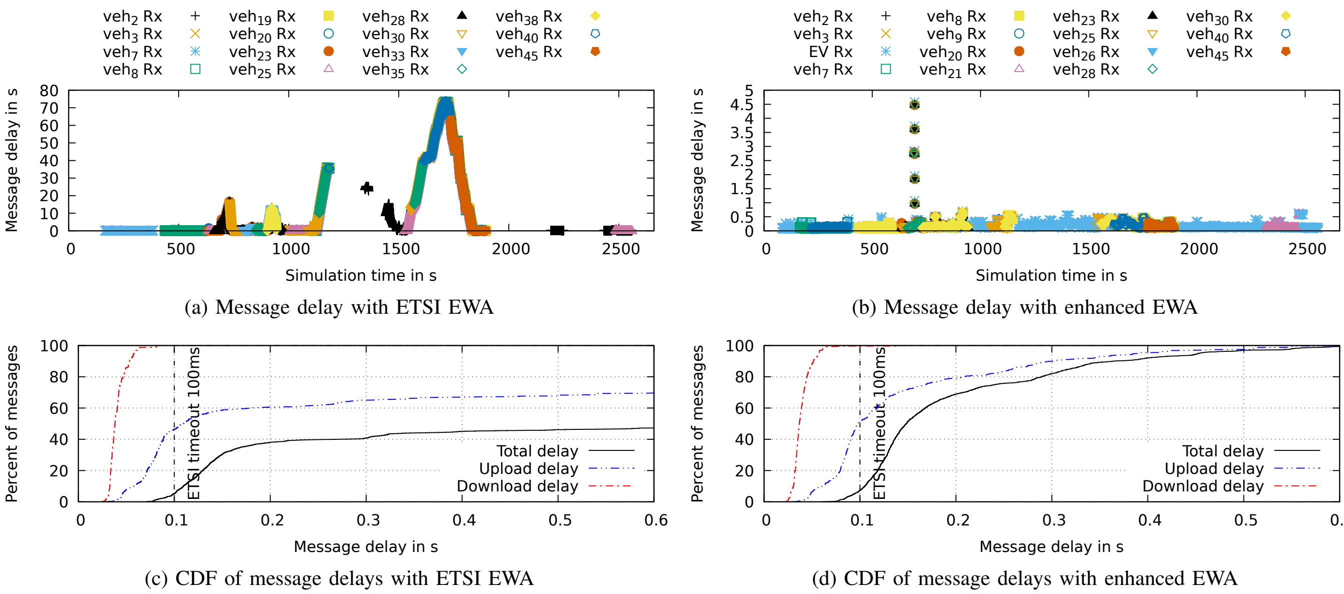
In total, we simulated 50 different vehicles equipped with EWA receivers over a total simulation duration of $1\,\mathrm{h}\ 0\,\mathrm{min}\ 55.9\,\mathrm{s}$ with the EWA sender assigned to the fifth vehicle, defining it as the emergency vehicle.

VII. Evaluation

In this section we first analyze the simulation results gained by using the ETSI EWA. In a second step we compare these results with the simulation outcome of our enhanced EWA version and show that it is beneficial for cellular network usage. As the ETSI defined an upper bound for message delays of $100\,\mathrm{ms}$ we focus the evaluation on packet delays.

For the ETSI EWA simulation Figure 7a shows a plot of the combined delays of DENMs sent from the emergency vehicle to the geoserver and DENMs forwarded from the geoserver to the vehicles within the broadcast area against the simulated time. The combined delays peak at different levels with the highest peak at over $70\,\mathrm{s}$ and others still being greater than $10\,\mathrm{s}$. These peaks can be traced back to intervals with little available upload data rate in the emergency vehicle's cell sector. There congestion occurs as packet queues are filled and cannot be depleted fast enough, due to insufficient network capacities. The visible gaps in this plot display time intervals where no vehicles were in the emergency vehicles' broadcast area and thus no DENMs were forwarded. Figure 7c shows the dispersion of message delays by displaying the upload delay, the download delays and the combined delays, using a *Cumulative Distribution Function* (CDF). Obviously, the downstream delay of the simulated messages is within the usual range of 3G networks. But the simulated upstream delays are high for a large amount of the received messages.

Figure 7b shows the combined message delays for the enhanced EWA, using the modifications described in Section VI-C. It should be noted that our enhanced EWA uses CAMs instead of DENMs in the upstream direction (from the emergency vehicle to the geoserver). In the downstream direction (from the geoserver to the vehicles) DENMs are used as before, except that they are also sent to the emergency vehicle. Also, the frequency at which CAMs and DENMs are sent is now at $1\,\mathrm{Hz}$ for both message types, lowering the network load in both the upload and the download. There are no more gaps in this plot as the emergency vehicle now receives DENMs, too, so there is a continuous amount of received messages pictured. As expected, the combined message delays show peaks, that are much lower than before. Thus, the scale of the y-axis is different from Figure 7a. The CDFs in Figure 7d show that the downstream delays did not change significantly. As expected, the reduction of the upstream traffic is beneficial for the upstream delays and thus for the combined message delays. Furthermore it is obvious that the upper bound of $100\,\mathrm{ms}$ for the message delay has to be raised for cellular network usage, as the mobile cellular



(a) Message delay with ETSI EWA

(b) Message delay with enhanced EWA

(c) CDF of message delays with ETSI EWA

(d) CDF of message delays with enhanced EWA

Figure 7. Message delays and CDFs of message delays for the ETSI EWA and enhanced EWA.

network has to be traversed two times until a messages reaches vehicles in the broadcast area.

The direct comparison of both simulations shows that the simulated 50 vehicles never over-saturated the downstream in the simulated area. But it also shows that the enhanced EWA largely reduced congestion in the upstream direction. This leads to a combined message delay of less than 600 ms for more than 99 % of the messages for the enhanced EWA, while only 48 % of the messages reached their destinations in the same time span using the ETSI EWA. It illustrates how important a thorough analysis and modification of parameters defined for 802.11p V2X applications is, when applications are ported to use with cellular networks. It also underlines that trace-based simulation can significantly speed up this process.

VIII. Related Work

This section gives an overview on related work in the fields of coupled simulators suitable for V2X simulation, trace-based network simulation and cellular network simulation.

A. Coupling of Traffic and Network Simulation

1) Mixed Simulation *(MiXiM):* MiXiM [14] is a combination of various simulation models and frameworks in OMNeT++. It allows the simulation of mobile wireless networks by an in-depth simulation of all layers of 802.11 networks. It uses the *Mobility Framework* [15] for node movements with predefined or random routes. MiXiM does not offer an interface to couple it with any other simulator.

2) Vehicles in Network Simulation *(Veins):* Veins [16] combines OMNeT++ and MiXiM with the road traffic simulator SUMO. It provides a bidirectional coupling allowing feedback and reactions on simulated events. Veins uses the *Traffic Control Interface* (TraCI) [17] as the control protocol, which offers an interface to all available SUMO values and allows changes of some values. It can be used to modify the traffic simulation, by e. g. requesting a vehicle to brake or change its route, based on simulated network events. This enables Veins to integrate vehicular traffic data and send feedback to SUMO for possible changes in how the vehicles behave, but restricts its use to only a bidirectional coupling between SUMO and itself. Veins is not easily extendable to include other simulators, because no general standardized protocol for communication between simulators is defined.

3) V2X Simulation Runtime Infrastructure *(VSimRTI):* VSimRTI [2] introduced a loosely multi-directional coupled framework for a unified simulation of different V2X features. Thereby the VSimRTI core is bidirectionally coupled with each simulator using a *High Level Architecture* (HLA) [18] formed by a *federate* and an *ambassador* per simulator. The *ambassador* of a simulator defines which messages the simulator is interested in and which messages it can provide. Thus, at the arrival of a message, the VSimRTI core forwards it to the ambassadors of all interested simulators. An ambassador receiving a message forwards it to the corresponding *federate*, which translates it to the “language” of the simulator. By using this open interface, one can add new simulators to VSimRTI or exchange simulators, like e. g. switch network simulators. Further on, VSimRTI already offers a full fledged V2X application simulator. We chose VSimRTI for our simulations as it offers the broadest range of supported simulators, is easily extendable and already offers a V2X application simulator.

B. Trace-based Wireless Network Simulation

One of the earliest and most inspiring papers on trace-based simulation of mobile networks is [19]. The authors wanted a reproducible simulation of their *WaveLan*, which is a predecessor of todays *IEEE* 802.11 WIFI. Therefore they conducted measurements in their network, sampling the bandwidth, the delay and the loss probability for discrete periods of time. Afterwards they derived a simple network model based on their traces. During their modeling, the authors assumed symmetric delays. But later on they discovered that this assumption did not hold. To overcome this, they suggested to use synchronized, high precision clocks.

C. Cellular Network Simulation

1) VSimRTI_Cell: VSimRTI's integrated mobile cellular network simulator VSimRTI_Cell contributes a probabilistic simulation based on three mathematical simulation models. In VSimRTI version $0.14.0$, VSimRTI_Cell offers a configurable delay simulation. Based on *delay regions* and a given *delay type* (constant delay, random delay with simple and gamma distributions, and a random delay with a gamma distribution including the current node speed), the network's transport delay is calculated. VSimRTI_Cell acts on probabilistic models, thus the simulation area has to be measured and divided into delay regions. Depending on the size of delay regions and quality of measured or estimated data, the simulation outcome might differ significantly from reality.

2) Trace Based UMTS Simulation *(TBUS):* Introduced in [8], TBUS forms the base for the simulation model we used here. A verified implementation of this simulation model already exists and has been used for simulations presented in the defining paper. TBUS simulates mobile cellular network conditions using real world network characteristics. The simulation model and the cell share extension introduced in this paper are explained in Section IV. By using measurements of mobile cellular networks as a simulation basis instead of probabilistic or mathematical models, TBUS realistically simulates the traversal of packets through the cellular network.

IX. Conclusion

In this paper, we introduced a complete simulation tool chain allowing realistic simulation of V2X applications using mobile cellular networks. Therefore, it uses well established simulators in conjunction with an OMNeT++ implementation of our trace-based simulation model. Furthermore, we explained how the network measurements are conducted and post-processed to build the graph-based simulation database. Finally, we demonstrated the usability of our tool chain by simulating ETSI's Emergency Warning Application, which was defined having 802.11p in mind, along with a cellular network optimized Emergency Warning Application. The comparison of the simulation results shows that the enhanced EWA version performs much better in the exemplary simulation area, allowing 99% of all messages to arrive in less than $600\,\mathrm{ms}$. While the presented simulation only covers one single area and thus is not universally valid, we are confident that the simulation results closely match what a similar field test in the same area at the same time would have shown.

For the future, we plan to further enhance the introduced cell share model. And, in cooperation with Fraunhofer FOKUS, we plan to incorporate the changes we made to VSimRTI into the next VSimRTI release. Furthermore, the impact of the utilized measurement traffic on the simulation of diverging V2X communication patterns has to be evaluated.

References

[1] simTD Konsortium, "simTD: Sichere Intelligente Mobilität Testfeld Deutschland," [Accessed Sep. 12th 2015]. [Online]. Available: http://www.simtd.de

[2] B. Schünemann, "V2X simulation runtime infrastructure VSimRTI: An assessment tool to design smart traffic management systems," *Computer Networks*, vol. 55, no. 14, pp. 3189 – 3198, 2011.

[3] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "SUMO – Simulation of Urban MObility," in *The Third International Conference on Advances in System Simulation (SIMUL 2011), Barcelona, Spain*, 2011.

[4] A. Varga *et al.*, "The OMNeT++ discrete event simulation system," in *Proceedings of the European simulation multiconference (ESM'2001)*, vol. 9, no. S 185. sn, 2001, p. 65.

[5] N. Goebel, T. Krauthoff, K. Graffi, and M. Mauve, "Moving Measurements: Measuring Network Characteristics of Mobile Cellular Networks on the Move," 2016, manuscript submitted for publication.

[6] OpenStreetMap, "OpenStreetMap," [Accessed Feb. 14th 2016], Copyright and License: www.openstreetmap.org/copyright. [Online]. Available: http://www.openstreetmap.org/

[7] F. Marchal, J. Hackney, and K. W. Axhausen, "Efficient Map-Matching of Large GPS Data Sets - Tests on a Speed Monitoring Experiment in Zurich, volume 244 of Arbeitsbericht Verkehrs und Raumplanung," Tech. Rep., 2004.

[8] N. Goebel, M. Koegel, M. Mauve, and K. Graffi, "Trace-based Simulation of C2X-Communication using Cellular Networks," in *11th Annual Conference on Wireless On-demand Network Systems and Services (WONS 2014) - Special Session on VANETs and ITS3*, Apr. 2014, pp. 108–115.

[9] "Intelligent Transport Systems (ITS); Framework for Public Mobile Networks in Cooperative ITS (C-ITS)," European Telecommunications Standards Institute, Technical Report ETSI TR 102 962 V1.1.1, Feb. 2012.

[10] "Intelligent Transport Systems (ITS); Vehicular Communications; Geographical Area Definition," European Telecommunications Standards Institute, European Standard ETSI EN 302 931 V1.1.1, Jul. 2011.

[11] "Intelligent Transport Systems (ITS); Vehicular communications; GeoNetworking; Part 4: Geographical addressing and forwarding for point-to-point and point-to-multipoint communications; Sub-part 1: Media-Independent Functionality," European Telecommunications Standards Institute, Technical Specification ETSI TS 102 636-4-1 V1.1.1, Jun. 2011.

[12] "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service," European Telecommunications Standards Institute, Technical Specification ETSI TS 102 637-2 V1.2.1, Mar. 2011.

[13] "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service," European Telecommunications Standards Institute, Technical Specification ETSI TS 102 637-3 V1.1.1, Sep. 2010.

[14] A. Köpke, M. Swigulski, K. Wessel, D. Willkomm, P. T. K. Haneveld, T. E. V. Parker, O. W. Visser, H. S. Lichte, and S. Valentin, "Simulating Wireless and Mobile Networks in OMNeT++ the MiXiM Vision," in *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops*, ser. Simutools '08. ICST, Brussels, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, pp. 71:1–71:8.

[15] "Mobilty Framework for OMNeT++," [Accessed Aug. 9th 2015]. [Online]. Available: http://mobility-fw.sourceforge.net

[16] 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, pp. 3–15, Jan. 2011.

[17] A. Wegener, M. Piórkowski, M. Raya, H. Hellbrück, S. Fischer, and J.-P. Hubaux, "TraCI: An Interface for Coupling Road Traffic and Network Simulators," in *Proceedings of the* 11^{th} *Communications and Networking Simulation Symposium*, ser. CNS '08. New York, NY, USA: ACM, Apr. 2008, pp. 155–163.

[18] F. Kuhl, J. Dahmann, and R. Weatherly, *Creating Computer Simulation Systems: An Introduction to the High Level Architecture*. Prentice Hall PTR, 2000.

[19] B. D. Noble, M. Satyanarayanan, G. T. Nguyen, and R. H. Katz, "Trace-based mobile network emulation," *SIGCOMM Comput. Commun. Rev.*, vol. 27, no. 4, pp. 51–61, 1997.