FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Department Informatik

Technical Reports / ISSN 2191-5008

Anatoli Djanatliev, Kai-Steffen Hielscher, Christoph Sommer, David Eckhoff, and Reinhard German (Eds.)

Proceedings of the 5th GI/ITG KuVS Fachgespräch Inter-Vehicle Communication (FG-IVC 2017)

Technical Report CS-2017-03

April 6th-7th, 2017

Please cite as:
Anatoli Djanatliev, Kai-Steffen Hielscher, Christoph Sommer, David Eckhoff, and Reinhard German (Eds.), “Proceedings of the 5th GI/ITG KuVS Fachgespräch Inter-Vehicle Communication (FG-IVC 2017),” Friedrich-Alexander-Universität Erlangen-Nürnberg, Dept. of Computer Science, Technical Reports, CS-2017-03, April 6th-7th, 2017.

Friedrich-Alexander-Universität Erlangen-Nürnberg
Department Informatik

Martensstr. 3 · 91058 Erlangen · Germany

www.cs.fau.de

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Proceedings of the 5th GI/ITG KuVS Fachgespräch Inter-Vehicle Communication (FG-IVC 2017)

April 6-7, 2017 in Erlangen

KuVS

GI ITG

General Chair
Anatoli Djanatliev

Organisation Co-Chairs
Kai-Steffen Hielscher
Reinhard German

Program Co-Chairs
Christoph Sommer
David Eckhoff

Event website: http://www7.cs.fau.de/fgivc2017

Contents

Proceedings of the 5th GI/ITG KuVS Fachgespräch Inter-Vehicle Communication (FG-IVC 2017)
Anatoli Djanatliev, Kai-Steffen Hielscher, Christoph Sommer, David Eckhoff, and Reinhard German (Eds.)

Methodological Challenges of the Evaluation of Intelligent Transportation Systems by Simulation

Holger Döbler and Stefan Dietzel
Humboldt-Universität zu Berlin, Berlin, Germany
Email: holger.doebler@informatik.hu-berlin.de, stefan.dietzel@hu-berlin.de

***Abstract*—Users of road networks are often assumed to choose their routes according to Wardrop's first principle, which leads to a user equilibrium. However, by coordination of route choice, the users may significantly reduce their average travel cost, e. g., travel time. This potential for improvement is an obvious application for future intelligent transportation systems (ITS). In this article we discuss the various explicit and tacit assumptions one has to make in order to study the characteristics of cooperative route choice methods, e. g., assumptions about the traffic model, drivers' behavior, available information, and communication channels. We motivate our discussion with an exemplary study of the effects of limited market penetration of a simple idealized toy ITS that optimizes its users' total travel cost by coordinating their route choice using simulated annealing.**

I. Introduction

To motivate the potential benefits of cooperative route choice in urban road networks, consider the prisoner's dilemma [1] or Braess' paradox [2]. The former is a simple two player game often studied in game theory that demonstrates that cooperative behavior may be beneficial for all players, compared to egoistic behavior. Braess' paradox is an example of a congestion game that demonstrates how egoistic users of an idealized road network may benefit from closing one link in the network or, equivalently, from cooperatively agreeing not to use that link at all.

It has often been proposed [3], [4] that cooperative route choice is a potential application of intelligent transportation systems (ITSs), where users share their knowledge and transport demand in order to compute a globally optimal route assignment or at least a route assignment that is better than the user equilibrium. The problem for system optimal route assignment has been solved macroscopically for static flows in small example networks [5], but macroscopic models are usually not applicable to time-varying demand. There also exist first approaches to compute approximations of optimal cooperative routes for time-varying demand using stochastic optimization and microscopic/mesoscopic traffic models [6]–[8] but their authors focus mainly on the algorithmic challenge to compute the route assignment efficiently and robustly.

Instead of developing a complex route choice optimization scheme with all bells and whistles, we address the methodological boundary conditions that have to be applied when analyzing the performance and achievements of the optimization. As an example we present an idealized ITS that computes an approximative optimal route assignment based on perfect knowledge about future demand and full user compliance. To evaluate this optimization scheme we compare the resulting total travel time of all users in the optimized case with the dynamic user equilibrium (DUE). In order to make the evaluation more interesting, we assume that the ITS is not used by all drivers in the road network but only by a fraction $r \in [0, 1]$ of them. This is, we want to study the ITS's performance depending on its market penetration.

The purpose of introducing this ITS is, however, solely to have an example of an ITS' evaluation. This example contains many idealizations that are commonly made when when studying traffic optimization by means of simulation. In the following we discuss these idealizations' impact on the evaluation's results and address how the idealized assumptions may be loosened in order to make the evaluation's results more practically relevant.

II. Toy ITS

To study the effects of route choice optimization we need a traffic model/scenario as well as an optimization scheme. On the one hand we would like to study a *real world* traffic scenario with realistic demand instead of a constructed minimal example network like the Braess' paradox' graph because constructed examples tend to carry the risk of being pathological examples. On the other hand our primary goal is to reflect on the evaluation. Therefore neither does the traffic model need to be very accurate, nor does the optimization scheme need to be efficient. Instead we stick to a simple traffic model as well as a simple optimization strategy in order to avoid the accidental introduction of unforeseen side effects or biases.

We propose a toy ITS that optimizes the total travel time of all of its users by adapting their routes. It is laid out as pure offline optimization, i. e., the complete origin-destination-matrix (OD-matrix) describing the demand of all drivers is assumed to be known a priori as a list of tuples <departure time,origin,destination>. In addition, the demand and fixed route choice of all other drivers in the network, the *non-users*, is assumed to be known as a list of tuples <departure time, route>. Note that throughout this article we use the term *user* for drivers using the ITS, *non-user* for egoistic drivers, and *drivers* to refer to all of them. The objective function $F : \langle\text{routes}\rangle \mapsto \sum_{i\in\{\text{cars}\}} \mathrm{TT}_i$ is evaluated via the SUMO traffic simulator in mesoscopic mode MESO. An approximation of the resulting system optimal route assignment is computed iteratively using a rudimentary simulated

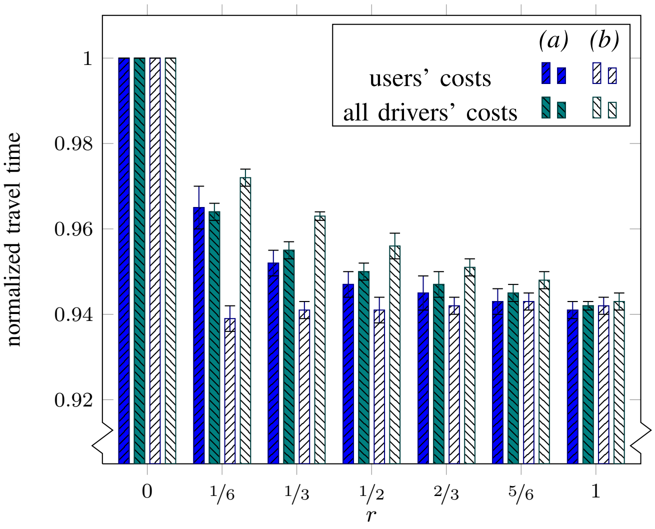


Fig. 1. Travel time saving of [users|all drivers] achieved by [*(a)* fully|*(b)* parochially] altruistic route choice behavior for different market penetration ratios r. *Before aggregation*, values were normalized to the travel times observed in case of the DUE. Each data point corresponds to ten measurements consisting of a user equilibration followed by optimization. Data for $r = 0$ was not measured but is given only for sake of completeness. The error bars depict the standard deviation.

annealing approach. In each iteration a solution candidate c_i is modified by changing a few users' routes resulting in a modified solution candidate c'_i. As choice set for each vehicle's routes we use the paths with highest local optimality ratio as described in [9]. $F(c'_i)$ is computed using MESO and depending on $F(c_i)$ and $F(c'_i)$ and a temperature T, either c_i or c'_i is selected as c_{i+1}.

As toy scenario we use the iTETRIS Bologna scenario [10] that incorporates 11,000 trips within one hour. We want to investigate a situation where a fraction $r \in [0, 1]$ of all car drivers use an ITS in order to cooperatively choose routes that minimize the total travel time. The other $(1 - r)$ of the users are assumed to stick to the DUE routes that they had used, if no one used the ITS.

We want to study and compare the following two cases: in case *a* users are truly altruistic in the sense that they seek to optimize the total travel time of all drivers, users and non-users. In case *b* users are parochially altruistic, so they optimize the total travel time of users but disregard the travel time of non-users.

In Fig. 1 we show the users' costs and all drivers' costs relative to the corresponding DUE costs for different values of ITS market penetration rate r. Data for altruistic user behavior, where the objective function is the sum of travel times of all drivers, is shown as well as data for parochially altruistic user behavior, where the objective function is the sum of users' travel times only, disregarding other drivers. The values for $r = 0$ correspond to all drivers behaving egoistically, i. e., to the pure DUE and therefore equal 1. $r = 1$ corresponds to full system optimal routing, so we can read from the plot that the DUE has about 6% higher costs than the computed system optimum. From the intermediate cases we read two rather surprising outcomes: first, the effect of optimization is large already for small market penetration ratios for both flavors of altruism. Second, and even more surprisingly, even in case of truly altruistic optimization, the users' benefit seems to be greater than the non-users benefit for optimization. This observation is counter-intuitive in so far that while users and non-users are equally respected in the objective function, the users are those potentially taking detours, while non-users stick to the unilaterally fastest routes. For the latter we see at least two possible explanations: on the one hand, the data for users' costs and total costs are very close, yet measured with a high uncertainty. Therefore this difference might be a random artifact due to the small sample size. On the other hand, we assumed the non-users to stick to their precomputed DUE routes. After optimization, however, the traffic conditions have changed and these routes to not obey Wardrop's first principle of an egoistic driver anymore. Therefore users are, by means of the optimization, not only able to take detours in order to allow for less congestion. Users, in contrast to non-users, are also able to adapt their routes to the altered traffic conditions.

This gives rise to the question whether we used a good model for egoistically driving non-users. We will come back to this question in Section III-C where we discuss the assumptions about the unoptimized *status quo* in more detail.

III. Methodological Issues

When we set up the scenario for our evaluation we made various assumptions. Some of these are now addressed in detail.

A. Input uncertainty

We assumed the OD matrix and the traffic model to be known exactly. In order to measure in how far our results depend on this knowledge, we introduce some noise in the OD matrix in terms of randomly perturbing the departure times. For given original departure times $(t_i) = (t_0, \ldots, t_N)$ that come with the iTETRIS scenario, we construct perturbed departure times $(t_i)'$ as $t'_i = t_i + \Delta t_i$ where the Δt_i are drawn uniformly at random from $[0, 2X]$ with the constraint that $\overline{\Delta t} = \frac{1}{N} \sum_{i=1}^{N} \Delta t_i = X$. We applied this perturbation with different values of $\overline{\Delta t}$ and repeated the optimization for $r = 1$ in two different ways (see Fig. 2). When we optimize routes for the original unperturbed departure times and evaluate the objective function for perturbed departure times, the resulting travel times show a surprising dependence on $\overline{\Delta t}$. For very small perturbations the travel times become higher than the corresponding DUE and have a large variance. This effect however decreases for larger perturbations of a few seconds per vehicle. For even larger perturbations in the order of tens of seconds per vehicle the travel time's variance stays low, while the mean value further increases. We explain that effect with the complex nature of the traffic model, especially in the vicinity of a system optimum, i. e., when bottlenecks are

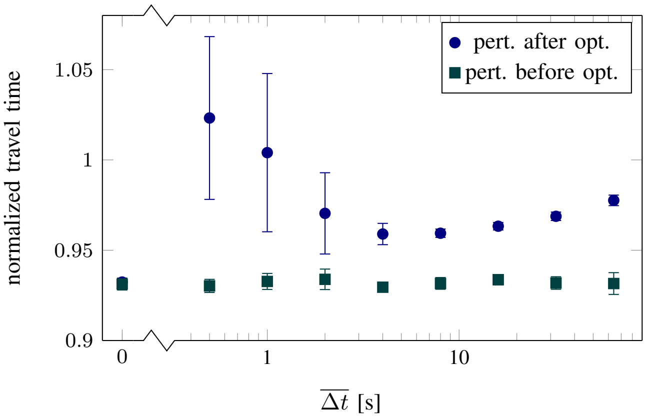


Fig. 2. Travel time savings of optimized routes ($r = 1$, altruistic) for perturbed departure times. For the upper data series a fixed optimized route assignment was evaluated for different perturbations. For the lower data series, routes where optimized for each instance of departure time perturbations. All travel times are normalized to the DUE with unperturbed departure times.

loaded at the very edge of their capacity and the system is at the border to become congested.

If, on the other hand, we perturb the departure times first and then compute system optimal routes for the modified departure times, we retain the travel time savings to the full extent within the observed measurement uncertainties.

We conclude that our ITS indeed relies heavily on the exact knowledge of the OD matrix and is therefore not a suitable approach for real world applications. We go even further and postulate that the fragility of our solutions is founded in the requirement of pure offline optimization. An optimal route assignment for a noisy OD matrix can, in general, not be computed a priori but it will will be necessary to take additional information about the evolving traffic state and future traffic demand into account as it becomes available. This ansatz resembles *online* optimization.

Yet we emphasize that the non-applicability of offline optimization does not invalidate our results concerning truly vs. parochially altruistic behavior in limited marked penetration situations. As we have shown by computing the optima after applying the departure time noise, the potential for improvement exists, despite the algorithm to compute the route assignment was based on unrealistic assumptions.

In the following sections we highlight other questionable assumptions commonly made when evaluating system-optimal navigation algorithms. For the sake of brevity, we restrict ourselves to propose ways how to empirically study their impacts, instead of presenting more of these meta-evaluations in this article.

B. Quality of the traffic model

The traffic model we use in our evaluation to measure the benefits of our ITS (the assumed *real* model) is the very same model that we use in our optimization algorithm to estimate the value of the objective function. This implies the tacit assumption that our knowledge and implementation of the real traffic model is perfect.

To study the impact of this assumption one could either artificially modify the parameters of one of these models or even use completely different traffic models.

C. Dynamic user equilibrium

In our toy model we used SUMO's `duaIterate` tool that performs MESO simulations alternately with shortest path search to compute an approximation of the DUE. This DUE that we assume as the status quo of *unoptimized route choice* is based on the following assumptions, each of which is already somewhat questionable on its own: first, all drivers choose their routes egoistically according to Wardrop's first principle. Second, the characteristics of the road network as well as the daily traffic demand do not vary from day to day. Third, all drivers have perfect knowledge about the future traffic conditions, because of their knowledge of historical data.

It is argued [11], [12] that based on these assumptions the route choice of real world drivers in a daily recurring traffic scenario will iteratively converge towards a DUE.

So an interesting question is: "In how far is the 'omniscient homo economicus in a strictly periodic world' a good model of the average real urban driver?" While answering this question quantitatively would require some sociological field research, we could experiment with mixing some drivers with a fundamentally different strategies into our model and measure the effects. This question is also addressed in [13], [14].

D. Elastic demand

As already stated, the knowledge on future demand in a real world traffic scenario is limited to some uncertainty. Demand may fluctuate between successive days or follow trends on different time scales but is still assumed to be an independent input to the problem.

But the assumption of a rational driver imposes even more complex effects to take place. Consider a commuter who wants her travel demand satisfied with as low cost (travel time) as possible. Under the presence of public transport she might regularly use the train because all reasonable road routes are always jammed and therefore induce a higher cost. If the road traffic is optimized using an ITS, her shortest road travel time would drop below the train's travel time which would in turn let her use the car, thereby increasing the total traffic demand on the road network. Thus, real world traffic demand will depend on the traffic conditions and thereby on the route choice as well.

Again, we know of no quantitative model of how an average traveler considers mode change as an option in order to achieve minimal cost.

On the other hand it would be fairly simple to experiment with such drivers, user or non-user, in scenarios like iTETRIS Bologna, that already contain public transport.

E. Required communication channels

Already the DUE requires some communication because of the assumption that every driver has perfect knowledge about historical data. On the other hand one could argue that each driver gains the knowledge about historical traffic states purely from periodically probing different routes. At the same time there are already well established channels available for distribution of traffic information, like FM radio traffic reporting, Traffic Message Channel, or conventional proprietary car navigation systems. The drivers' usage of this offer of information will impact the characteristics of the resulting DUE, but to our knowledge there is no known quantitative model of the average egoistic urban driver with respect to how much and which specific information is taken into account to perform route choice.

An ITS may require even more communication. For the proposed pure offline optimization strategy it would be sufficient that each user publishes her desired departure time and location as well as destination location a priori to, e. g., a central traffic coordination system and receive a route in return. While this information could be transmitted with messages of a few bytes in either direction without any necessity for short delays, we have pointed out in Section III that pure offline optimization will in general lead to fragile optima. To complement the offline approach with online optimization in order to react on unforeseen future traffic conditions requires the users to exchange information about the observed traffic state as well as to receive information about possibly changing routes during the trip. This is likely to increase the amount of data that needs to be transmitted between vehicles, either directly or via a centralized route coordination infrastructure. Because this information would need to be distributed in real time, it also manifests some upper bound on acceptable communication delays.

In addition, by asking users to publish their traffic demands, we silently assume that all users know their own demand significantly ahead of time, that users have no privacy concerns regarding their trips, and that no user or non-user has the incentive to inject incorrect demand information in order to exploit the ITS.

IV. Conclusion and Outlook

Using a simplistic ITS and an evaluation of the effects of limited market penetration we have emphasized some of the assumptions that are made in our evaluation. All of these are idealizations that help to make the evaluation by means of traffic simulation less complex.

For some properties of the system, quantitative knowledge about the corresponding real world behavior cannot be gained with mathematical techniques and traffic simulation only. On the other hand we have pointed out how the effect of relaxing these tight constraints could be studied computationally.

References

[1] A. Rapoport and A. M. Chammah, *Prisoner's Dilemma: A Study in Conflict and Cooperation*. University of Michigan press, 1965, vol. 165.

[2] D. Braess, “Über ein Paradoxon aus der Verkehrsplanung,” *Unternehmensforschung*, vol. 12, no. 1, pp. 258–268, 1968.

[3] J. L. Adler and V. J. Blue, “A cooperative multi-agent transportation management and route guidance system,” *Transportation Research Part C: Emerging Technologies*, vol. 10, no. 5–6, pp. 433–454, Oct. 2002.

[4] T. Roughgarden, *Selfish Routing and the Price of Anarchy*. MIT press Cambridge, 2005, vol. 174.

[5] O. Jahn, R. H. Möhring, A. S. Schulz, and N. E. Stier-Moses, “System-Optimal Routing of Traffic Flows with User Constraints in Networks with Congestion,” *Operations Research*, vol. 53, no. 4, pp. 600–616, 2005.

[6] Y. Shigehiro, T. Miyakawa, and T. Masuda, “Road traffic control based on genetic algorithm for reducing traffic congestion,” *Electronics and Communications in Japan*, vol. 95, no. 4, pp. 11–19, Mar. 2012.

[7] J. Pan, M. A. Khan, I. S. Popa, K. Zeitouni, and C. Borcea, “Proactive Vehicle Re-routing Strategies for Congestion Avoidance,” in *2012 IEEE 8th International Conference on Distributed Computing in Sensor Systems*, May 2012, pp. 265–272.

[8] D. Cagara, B. Scheuermann, and A. L. C. Bazzan, “Traffic optimization on Islands,” in *2015 IEEE Vehicular Networking Conference (VNC)*, Dec. 2015, pp. 175–182.

[9] H. Döbler and B. Scheuermann, “On Computation and Application of k Most Locally-Optimal Paths in Road Networks,” in *Fachgespräch Inter-Vehicle Communication 2016*. Berlin, Germany: Humboldt-Universität zu Berlin, Mar. 2016, pp. 32–35.

[10] L. Bieker, D. Krajzewicz, A. Morra, C. Michelacci, and F. Cartolano, “Traffic Simulation for All: A Real World Traffic Scenario from the City of Bologna,” in *Proceedings of the SUMO2014 — Modeling Mobility with Open Data*. Deutsches Zentrum für Luft- und Raumfahrt e.V., May 2014, pp. 135–146.

[11] C. Gawron, “Computing User Equilibria in Large Street Networks,” Ph.D. dissertation, Universität zu Köln, 1998.

[12] J. Bie and H. K. Lo, “Stability and attraction domains of traffic equilibria in a day-to-day dynamical system formulation,” *Transportation Research Part B: Methodological*, vol. 44, no. 1, pp. 90–107, Jan. 2010.

[13] M. Sharma, J. K. Gupta, and A. Lala, “Survey of Route Choice Models in Transportation Networks,” in *Intelligent Computing, Networking, and Informatics*. Springer, New Delhi, 2014, pp. 1285–1290.

[14] E. Ben-Elia and E. Avineri, “Response to Travel Information: A Behavioural Review,” *Transport Reviews*, vol. 35, no. 3, pp. 352–377, May 2015.

Evaluating Misbehavior Detection for Vehicular Networks

Rens W. van der Heijden, Frank Kargl
Institute of Distributed Systems
Ulm University
Ulm, Germany
Email: rens.vanderheijden@uni-ulm.de

***Abstract*—The literature proposes a large variety of different misbehavior detection mechanisms for vehicular networks, which are designed to separate attacks and faulty data from legitimate data. Most of these mechanisms are evaluated using techniques adopted from the field of intrusion detection. However, because misbehavior detection is content-oriented, and includes the detection of faulty data, it is possible that such data may be indistinguishable from an attack. This paper discusses and critically reflects upon different evaluation strategies used in the literature, and provides some recommendations for authors.**

I. Introduction

Vehicular networks are one area where misbehavior detection provides a significant benefit, because integrity is more important than confidentiality, and the data correctness plays a much more significant role than for regular networks. The literature has already developed many schemes, which are evaluated in widely different ways and with very different tools. Some are analytical, some are simulative and sometimes there are limited experimental cases.

Similarly, the data that is verified is normally application-specific, and not independent of surrounding vehicles (i.e., a ground truth cannot be established for just the particular message content that is being evaluated), which makes analytical approaches very difficult. Additionally, analytical approaches often require simplifications that could again be exploited by an attacker. Experimental approaches have different issues; experiments are expensive to perform, and cannot necessarily be applied to all settings, especially when the consequences are dangerous (e.g., attempting to trigger a crash on a highway for experimentation is not ethically or financially feasible). Nevertheless, experimental approaches can still provide some benefit in estimating the behavior of a system in an attacker-free environment, and the resulting data set can be artificially extended with attacks [1].

Work on Intrusion Detection System (IDS) validation has already seen that it is difficult to compare results between IDSs, but this is made harder by the fact that Misbehavior Detection Systems (MDSs) are much more varied, and often designed to work with data. Many other questions play a role: should detection happen on a per-packet basis or a per-node basis? Is the goal to eventually detect all attackers, or to detect the most attackers with the lowest latency? What kind of attacks are considered? Is there application logic that tries to be fault tolerant?

The evaluation of misbehavior detection mechanisms is often strongly influenced by what kind of attacks the authors were considering. This makes comparisons across published results very difficult. There seem to be no satisfactory evaluation metrics in the literature that both catch the subtleties of misbehavior detection, while still remaining general enough to be applicable to more than just a specific mechanism.

In this paper, we aim to give an overview of the most common evaluation techniques and metrics, and discuss their suitability for evaluating misbehavior detection. The contribution is two-fold; the paper can be used as reference material, but we also aim to provide a common ground that authors can use to compare their work to that of other authors.

II. Evaluation Strategy

In this paper, we limit ourselves to simulated evaluation strategies. These strategies enable reliable and reproducible experiments in a variety of settings, and enable analysis of detector performance on a large scale. Although there has been work on empirical analysis of attacks [1], such studies are extremely difficult to generalize and verify without a significant deployment of vehicular communication systems, which are still in their deployment phases at the time of writing. There is a wide variety of network and vehicle behavior simulation tools, with different scenarios; however, these are not the primary topic of this paper. For more information on this topic, refer to studies that compare simulation environments [2], [3].

However, what is important to consider for this paper is the significant variables that could have an impact on overall detection performance, and which should be analyzed individually to make general statements about the suitability of a detection mechanism. The most significant parameters here include the amount of attackers, the amount of legitimate vehicles and the driving scenario. By studying these parameters, authors can gain an insight into whether their detector has weaknesses in specific scenarios, and how it performs when significant message loss is encountered, and these are often also used in the evaluation of vehicular applications. In all cases, repetitions under different simulation seeds are used whenever probabilistic models of, e.g., the communication channel are considered. The amount of attackers is a first step to determine whether the detector is resilient against attacks.

Resilience against attack is a metric that is difficult to quantify, however; it is particularly challenging to consider sufficiently distinct attacks, and to implement these attacks correctly. Because detectors are often designed to detect specific attacks, it is tempting to implement exactly these attacks and then show that the detector performs well. Therefore, it is important to clearly specify the attacker model, including how exactly this attack works, and whether or not attackers can cooperate (collaborative attackers) and whether or not attackers can create a limited amount of additional identities (Sybil attackers). In the latter case, the attacker uses different pseudonymous identities that are possibly linkable by an authority; in the former case, multiple independent vehicles are attacker-controlled (e.g., this could be due to malware).

Having considered the evaluation strategies, authors should next consider what an appropriate metric for validation is. Most mechanisms will perform well in some metrics, and worse in others; it is therefore important to clearly state what the evaluation metric is. For comparative studies, this means that different metrics need to be considered, requiring an evaluation strategy that is as independent from the evaluated mechanisms as possible.

III. Evaluation Metrics

Almost all papers discussing detectors evaluate these through some measure of detection quality. However, there are many different strategies and nuances to this process, e.g., how it is computed and how it is aggregated over different messages and vehicles. In addition to detection quality, some alternative metrics exist, e.g., how long it takes for an attacker to be detected, or the message overhead (if any); we discuss these metrics separately.

A. Confusion Matrix

The most obvious approach to evaluate the quality of a detector's results is to use the confusion matrix, which is also used in other fields (e.g., medicine and machine learning). This matrix describes various combinations of false positives, false negatives, true positives and true negatives and their rates with respect to the entire population. This includes metrics like accuracy, which is the amount of correctly classified events divided by the total population, and false omission rate, which is the amount of false negatives in the set of all negatives. Most papers in the area of misbehavior detection that use this type of metric use the false negative rate to quantify the risk of missing detections, and the false positive rate as the risk of an incorrect detection event. Opinions differ widely on what acceptable values are (e.g., for intrusion detection in networks with a lot of traffic, a false positive rate over 0.001 is considered very bad), and it is difficult to make general statements about this, because the impact of a false detection event is significant.

However, it is actually not trivial to classify a detectors' results into these categories when evaluating the detector in a simulation. Notably, in distributed detection scenarios such as vehicular networks, proximity to the attacker plays a significant role in how likely a vehicle is able to detect an attack. Not all nodes in the network will hear all messages, so if the data is aggregated across different detectors, one must take care to normalize these results: instances of a detector do not necessarily produce the same output for a given message, and not every message is seen equally often. Simple normalization may not be sufficient here: if a specific subset of receivers produces very high false negatives, while the aggregate of all receivers on average performs very well, this could still mean the detector is bad. In other words, aggregating and normalizing over an entire simulation may hide the fact that there is a weakness in a specific scenario.

Having established a way to normalize and aggregate the detection results still leaves other questions open. The definition of the input of the detector is one of these factors: does the detector take a single message and output a classification, or does it take a stream of messages and output an eventual classification? In the latter setting, one needs additional metrics to establish the timeliness of the system (e.g., detection latency: how long does it take for a detector to correctly classify an attack, after this attack starts?).

For a data-centric setting, using the confusion matrix is particularly difficult, because of the fact that many attacks are impossible to distinguish from legitimate messages. This can happen in two cases: either the message was sent by an attacker, but follows the expected behavior of vehicles (i.e., it is not a malicious message), or the attack is so marginal that it has no impact (and is thus indistinguishable from expected behavior or sensor noise). For example, an attacker might transmit a beacon with a position a few centimeters from its' actual position. This led some authors to conclude that application-oriented evaluations may be more suitable: if the attacker cannot achieve a goal, because the impact of false data is too small, then clearly the detection mechanism is effective. The disadvantage of this strategy is that the evaluation depends not only on the simulation aspects and the attackers' implementation, but also on the application implementation.

For data-centric detection mechanisms that are based on consistency, i.e., they consider multiple data sources and detect inconsistencies, it is often implicitly assumed that previously received data is true, and only the incoming packet is classified as legitimate or malicious. However, this means that message order is particularly significant: whenever a malicious message arrives first, it may trigger an additional false positive, because the next legitimate message differs from the malicious message.

Many papers implicitly discuss that detectors should also perform revocation or response – they exclude specific messages from those that are received, in order to prevent errors in the application. Similarly, some detection algorithms are inherently incompatible with the idea of evaluating individual message on a sequential basis, because they perform some batch processing (e.g., classifying vehicles instead of messages [4]).

Some authors use a pre-classified set of messages, which is not necessarily data-centric (e.g., Grover et al. [4] use a set of 3101 legitimate and 1427 malicious samples, with several types of attacks). This is common in the field of machine learning, where classifiers are often tested using this type of approach. However, since most detection algorithms in VANETs have different inputs, it is difficult to find a conclusive set that considers these various inputs, as well as contain the necessarily distinct types of attacks (especially when reputation is considered, which can be built over time). In other words, this approach is only valid if each sample is well-defined, which is not the case in our more general setting.

B. Alternative Accuracy Metrics

This led authors to use application-specific or detector-specific evaluation strategies, in order to demonstrate specific strengths of individual detectors in comparison to others from the literature.

a) Application behavior metrics: in many VANET scenarios, the specific values transmitted by an attacker are not necessarily relevant, but what is of interest is whether a receiving vehicle makes incorrect decisions about the state of the world. Application metrics aim to capture this subtle concept into a concrete metric. These metrics are useful, because they can also consider errors from other (i.e., non-malicious) sources, and are independent of a deep understanding of the detection mechanism. However, they require an application implementation, which is bug-prone and makes attacker implementation more complex.

One specific category application metrics is that related to schemes that detect routing misbehavior. Because routing misbehavior is historically closely linked to evaluation of routing schemes, some authors use routing performance metrics (such as arrival rates, consumed bandwidth and similar metrics) and changes to vehicle mobility [5], [6].

Another class of application metrics is much closer to the data that many data-centric detection mechanisms analyze; for example, this includes collision avoidance applications [7], [8] and in-network aggregation [9]. A disadvantage of these strategies is that it is hard to use them as a baseline for other studies, because often they are very specific.

b) Detector specific metrics: Some detectors have known sources of potential errors, often inherent to their design; a common strategy to deal with this is to approach and analyze these issues specifically. This is particularly useful when it is very clear what kind of error sources are to be considered. For example, Bimeyer et al. [10] used this type of metric to also include GPS error as a potential source of additional false positives for their scheme to analyze these effects in detail.

Another example of such an approach would be an evaluation where the impact of the attacker on the analyzed variable is. This only works for continuous variables, such as position information, where a simple distance metric is available to estimate the error between real positions and falsified positions. Rather than looking at how well attacks are detected only, this strategy would measure the distance between accepted attacker-generated data that is accepted as valid, and use this as a metric for detection quality. This approach is also viable if error sources are considered for legitimate vehicles. A disadvantage of this approach is that it may only be suitable for specific classes of detectors (or at least, it may put other detectors at a disadvantage).

C. Other Metrics

There are many other types of metrics available to authors seeking to evaluate their detection mechanisms. Some notable examples include:

a) Detection Latency: the time required for an attack to be detected. The exact definition of this metric differs type of detector, but typically it is the time between the first malicious packet received and the first detected malicious packet. This is particularly significant to measure the impact of reputation abuse in trust-based node-centric detection mechanisms: if trustworthy vehicles transmit malicious information, the potential impact of an attack is large.

b) Computational Cost: although relatively uncommon for vehicular networks, more traditional benchmarks such as computational cost can also be used. Most authors only check whether reasonable estimates of locally received messages can be analyzed in reasonable time (which, using 100 vehicles in range transmitting at 10Hz, is at most a millisecond per message), but particularly with proposals that include multiple detectors, scalability could be an issue.

c) Financial Cost: traditional IDSs are commonly evaluated by examining the cost associated with response and the corresponding variants in the confusion matrix [11]. In the case of vehicular networks, this is a bit more complex: the cost of a false negative is difficult to estimate, and strongly dependent on the scenario (i.e., high speed collisions have much higher cost, even though the classification of the event is exactly the same). Estimating this cost is also ethically complicated, because human lives are included in this process, which suggests that such an evaluation requires more extensive knowledge on how to deal with this suitable (as done, e.g., in invasive medicine).

d) Stability & Usability: one factor that is often forgotten or stated as future work only is that the detection mechanism should be sufficiently stable, such that the users' experience with the system is good enough. This is important, because users' trust in the system is strongly dependent on their perception of the systems' reliability. If the system continually warns or makes changes in response to possible attacks, but has an overall higher performance than other systems, then users may still perceive that system as very unreliable. This is one point where node trustworthiness over time will perform significantly better.

IV. Recommendations

We recommend authors to follow the developments in the simulation community, and take note of extensive standardized scenarios, such as the LuST scenario [12] for traffic simulation. This also includes pro-actively porting source code that is under development to more recent versions of the simulation environment wherever possible: ideally code should always be based on the most recent stable version of the simulator available while the paper is under review. This allows authors to benefit from quality of life improvements in the simulation environment, but allows them to consider, e.g., newer channel models.

For reasons outlined above, we also recommend using a variety of different attack strategies, which ideally have very different goals. This ensures that the authors can describe the qualities, as well as the limitations of their detector, which in turn can help future authors decide whether new attacks against this detector may be possible. It is also important to define a baseline in addition to these attacks, which can be undertaken in various ways (and this depends strongly on what the authors actually designed), and if possible a simulation of potential non-malicious sources of error. The simplest example is GPS error: adding an error to a GPS coordinate is relatively simple, but it makes the results much more representative than an idealized perfect positioning system for each vehicle, where it just “knows” its’ position.

The authors should at least consider how they aggregate the detection results across messages, vehicles and simulation repetitions, as discussed above, and clearly specify their approach. If possible, we recommend developing the simulation in such a way that multiple aggregation approaches can be used; there may not be a one size fits all solution.

Finally, we recommend publishing source code, or at least making it available to other researchers on request. This enables reproducibility, one of the core principles of scientific research, which allows research in this area to make faster and more meaningful progress. It would also enable studies that analyze the behavior of a variety of algorithms in different settings a much more efficient and less error-prone process. Finally, the authors themselves benefit from this process, because it is easier to compare to the literature.

V. Conclusion

In this paper, we have discussed several evaluation approaches for misbehavior detection. After describing several challenges and briefly surveying existing solutions, we gave some concrete recommendations that should be useful for authors that are studying misbehavior detection mechanisms. We plan to use these recommendation in combination with a framework that we are developing, named Maat, to evaluate potential detection mechanisms. As part of this project, we are also looking to decouple the execution of the simulation and the detection mechanisms, which would enable us to provide a data set to the community in addition to publishing our results.

Acknowledgment

This work was supported in part by the Baden-Württemberg Stiftung gGmbH Stuttgart as part of the project IKT-05 AutoDetect of its IT security research programme.

VI. References

References

[1] N. Bissmeyer, K. H. Schroder, J. Petit, S. Mauthofer, and K. M. Bayarou, “Short paper: Experimental analysis of misbehavior detection and prevention in vanets,” in *Vehicular Networking Conference (VNC), 2013 IEEE*. IEEE, 2013, pp. 198–201.

[2] R. Stanica, E. Chaput, and A.-L. Beylot, “Simulation of vehicular ad-hoc networks: Challenges, review of tools and recommendations,” *Computer Networks*, vol. 55, no. 14, pp. 3179 – 3188, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128611001629

[3] C. Sommer, J. Hrri, F. Hrizi, B. Schnemann, and F. Dressler, “Simulation Tools and Techniques for Vehicular Communications and Applications,” in *Vehicular ad hoc Networks*, C. Campolo, A. Molinaro, and R. Scopigno, Eds. Springer International Publishing, 2015, pp. 365–392, dOI: 10.1007/978-3-319-15497-8_13. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-319-15497-8_13

[4] J. Grover, N. K. Prajapati, V. Laxmi, and M. S. Gaur, “Machine learning approach for multiple misbehavior detection in vanet,” in *Advances in Computing and Communications*, ser. Communications in Computer and Information Science, A. Abraham, J. L. Mauri, J. F. Buford, J. Suzuki, and S. M. Thampi, Eds. Springer Berlin Heidelberg, 2011, vol. 192, pp. 644–653.

[5] J. Hortelano, J. C. Ruiz, and P. Manzoni, “Evaluating the usefulness of watchdogs for intrusion detection in VANETs,” in *IEEE International Conference on Communications Workshops (ICC)*. IEEE, May 2010, pp. 1–5.

[6] T. Leinmüller, E. Schoch, F. Kargl, and C. Maihöfer, “Decentralized position verification in geographic ad hoc routing,” *Security and Communication Networks*, vol. 3, no. 4, pp. 289–302, Jul. 2010. [Online]. Available: http://doi.wiley.com/10.1002/sec.56

[7] J. Petit, M. Feiri, and F. Kargl, “Spoofed data detection in vanets using dynamic thresholds,” in *Vehicular Networking Conference (VNC)*. IEEE, nov. 2011, pp. 25 –32.

[8] T. H.-J. Kim, A. Studer, R. Dubey, X. Zhang, A. Perrig, F. Bai, B. Bellur, and A. Iyer, “VANET alert endorsement using multi-source filters,” in *Proceedings of the seventh ACM international workshop on VehiculAr InterNETworking (VANET)*. New York, NY, USA: ACM Press, 2010, p. 51. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1860058.1860067

[9] R. W. van der Heijden, S. Dietzel, and F. Kargl, “Sedya: secure dynamic aggregation in vanets,” in *Proceedings of the sixth ACM conference on Security and privacy in wireless and mobile networks*. ACM, 2013, pp. 131–142.

[10] N. Bißmeyer, C. Stresing, and K. M. Bayarou, “Intrusion Detection in VANETs Through Verification of Vehicle Movement Data,” in *Vehicular Networking Conference (VNC)*. IEEE, 2010, pp. 166–173.

[11] A. A. Cárdenas, J. S. Baras, and K. Seamon, “A framework for the evaluation of intrusion detection systems,” in *IEEE Symposium on Security and Privacy*. IEEE, 2006, pp. 15–pp.

[12] L. Codeca, R. Frank, and T. Engel, “Luxembourg sumo traffic (lust) scenario: 24 hours of mobility for vehicular networking research,” in *Vehicular Networking Conference (VNC), 2015 IEEE*. IEEE, 2015, pp. 1–8.

Proceedings of the 5th GI/ITG KuVS Fachgespräch Inter-Vehicle Communication (FG-IVC 2017)
Anatoli Djanatliev, Kai-Steffen Hielscher, Christoph Sommer, David Eckhoff, and Reinhard German (Eds.)

Evaluating IVC for Planning Fully Automated Driving Actions in Multi-Storey Car Parks

Christoph Sippl*, Florian Particke†, Reinhard German‡ and Anatoli Djanatliev‡
*Audi AG, Ingolstadt, Germany, Email: christoph.sippl@audi.de
†Information Technologies, Friedrich-Alexander University Erlangen-Nuremberg (FAU), Germany
‡Computer Networks and Communication Systems, Friedrich-Alexander University Erlangen-Nuremberg (FAU), Germany
†‡Email:{florian.particke, reinhard.german, anatoli.djanatliev}@fau.de

***Abstract*—In this contribution we propose a simulation setup for Inter-Vehicle Communication (IVC)-evaluations of fully automated vehicles in multi-storey car parks. The simulation system targets the question, how latency in collective perception and environment data exchange affects the planning of driving actions. In order to meet requirements and ensure usability, we are not only working on the simulation system itself, but also validating the setup with experimental scenarios. Here, we inspect how latency in vehicle processing chains may influence situation interpretation and decision making for future driving actions. The modelled scenarios are dealing with Field-of-View (FoV) use cases, collective perception, situation interpretation and decision making. By means of the scenarios a contrast can be made, when the availability of environment data at different points of time lead from normal driving actions to more immoderate actions like emergency breaking.**

I. Introduction

Over the past years Advanced Driver Assistance Systems (ADAS) and fully automated driving have become more and more important, not only for car manufacturers, but also for other stakeholders. Various car manufacturers comment and estimate the introduction of fully automated vehicles for the next decade [1]. As door opener, various application scenarios are discussed [2]. An exemplary scenario in this context is fully automated parking in multi-storey car parks. One of the main requirements for fully automated vehicles is safety and accident-free driving. Furthermore, another key requirement is comfortable movement, as studies show an increase of acceptance and trust in automated vehicles with human-like and looking forward driving behaviour [3], [4]. To meet these requirements and ensure automated driving, vehicles must have a large set on sensors to perceive other traffic participants and static objects within their Field-of-View (FoV). Sequentially, after environment perception, automated vehicles have to analyse and interpret their environment and then take a decision for future actions. In cases of sensor blindness, existing dynamic objects which are not in the FoV may influence situation interpretation and decision making for future driving actions.

To investigate FoV scenarios in car park environments, we are setting up a microscopic, distributed and real time traffic simulation. This contribution addresses a work in progress simulation setup for IVC and evaluating fully automated driving functions. As various studies deal with Vehicle-to-Everything (V2X) and load case studies, this work focuses on vehicle processing chains. The main question is, how environment data exchange using IVC influences situation interpretation and decision making for future driving actions.

This contribution is organized as follows. Section II evaluates the relevance of this work by discussing related work. Section III presents the prototypical simulation architecture and the experimental scenarios. Section IV concludes this paper and points out next steps and future work in the field of latency effects for situation interpretation and decision making.

II. Relevance of this Work

After environment perception, fully automated vehicles must be able to analyse and interpret the recognized environment to enable situation awareness (SA). Mica R. Endsley introduced the so called SA model [5] in 1995, which describes a theoretical model in dynamic human decision making. The SA model is divided into three levels, where Level 1 describes the perception of elements (environment) in the current situation, Level 2 the comprehension of the current situation, and Level 3 the projection (cognition of intention) of the future status. The SA model is followed by decision making and the performance of actions. Further, the SA model can be transferred to technical problems and was taken up in previous works in the fields of robotics (e.g. [6]), automation of vehicles (e.g. [7]) and driving studies (e.g. [8], [9]).

A main task in automated driving is to interpret the environment and take decisions for future driving actions. While the dynamically changing environment has to be under permanent observation, the previously made decisions and planned driving actions have to be adapted. To meet the main requirements for fully automated driving, accident-free movement in a comfortable way, the system must have as much knowledge about the environment as possible. This guarantees sufficient situation interpretation and decision making. Further, not only the integrity of the knowledge about the environment is necessary, but it is also essential that the information is available as soon as possible. Even the large set of vehicle sensors has a limited view on the environment, as static or dynamic objects are too far away or hidden by other objects. Especially in urban traffic or closed environments like car parks, relevant elements are often not in the FoV of the automated vehicle. Thus, the process of situation interpretation and decision making is insufficient.

To prevent such circumstances, collective perception and environment information exchange using V2X communication was taken up by several publications and projects such as *Ko-FAS*[1] and *AutoNet2030*[2]. They are dealing with collective perception message formats, protocols and fusion architectures for cooperative perception [10-13]. Günther et al. presented the Environment Perception Message (EPM) format and highlighted the benefits of using collective perception in ADAS and fully automated driving by evaluating an obstacle avoidance scenario on a race track [14]. Another work authored by Günther et al. presents collective perception and Decentralized Congestion Control (DCC) in a simulation study [15] using the *Artery*[3] framework [16] and the *SUMO* framework [17], which has been coupled with the *Veins* framework [18].

As information exchange using V2X communication was discussed extensively in several works from the technical point of view, we are focusing on the question, what collective perception and latency mean for situation interpretation. Additionally, we are evaluating internal chains of the perception, decision making and action performing process.

III. Simulation Setup

A. Distributed Simulation Architecture

In order to get an expandable framework with a practical orientation, the simulation framework is built up as a distributed system including various tools (cf. Fig. 1). We are using the *Automotive Data and Time-Triggered Framework* (ADTF)[4] for implementing and providing prototypical vehicle functions. Depending on the scenario, every vehicle can be represented as a separate ADTF instance. The traffic simulation is realized using *Virtual Test Drive* (VTD)[5]. VTD allows to simulate traffic density and features in urban environments like traffic lights and foreign vehicles with realistic autonomous behaviour. Realistic pedestrian behaviour models are provided by integrating the pedestrian simulation framework *MomenTUMv2* [19]. ADTF and *MomenTUMv2* has been linked to VTD which, dependent on the use case, controls dynamic objects or only provides the basis for traffic simulation and visualization. Furthermore, VTD provides modules for setting up scenarios, editing scenarios and designing road networks. The multi-storey car park is realized in an independent implementation as an Message-Oriented Middleware (MOM). The car park is communicating with vehicles and other devices as described in the following subsection (cf. III-B).

B. Communication Architecture

The communication between the participating objects (vehicles and car park server) is realized using LTE-V and transmitting GeoNetworking (GN6ASL) packages (cf. Fig. 2). An

[1]http://www.kofas.de/ (Accessed: 01-03-17)
[2]http://www.autonet2030.eu/ (Accessed: 01-03-17)
[3]ETSI ITS-G5 application layer for *Veins*
[4]https://www.elektrobit.com/products/eb-assist/adtf/ (Accessed: 01-03-17)
[5]https://www.vires.com/ (Accessed: 01-03-17)

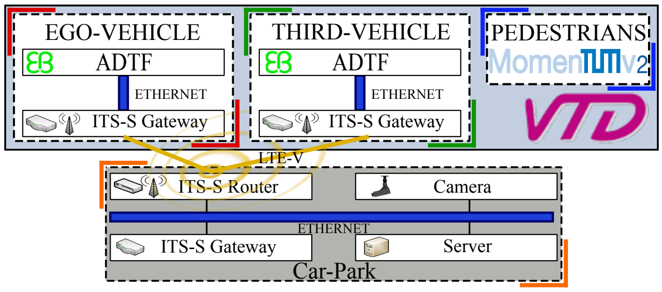


Fig. 1. Illustration of the expandable simulation architecture. According to the considered scenario, the vehicles (red and green highlighting) and the pedestrians (blue highlighting) are simulated as separate instances and linked to VTD or simulated in VTD. The car park (orange highlighting) is an independent message broker implementation.

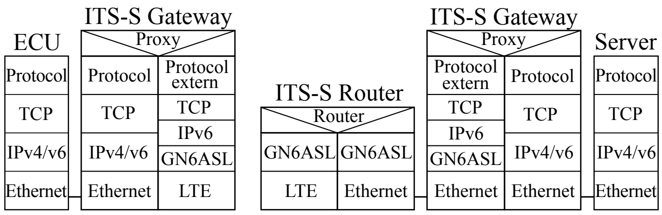


Fig. 2. Communication layer of the vehicle and multi-storey car park. The vehicle ECU is connected via Ethernet with the on-board ITS-S Gateway, which is communicating with further devices using LTE-V. The car park server is connected with an ITS-S Gateway via Ethernet. The Gateway is using a ITS-S Router for sharing environment information.

ITS-S Gateway on each object using a proxy service to convert a specific car park protocol into an external presentation and then sends the data using TCP/IPv6 and GN6ASL packages over LTE-V. On the car park server side, several ITS-S router send or receive the GN6ASL packages and then forward them using Ethernet to the ITS-S Gateway. The routers only bridge the different communication technology and have no further function implemented, but ensure large-scale radio coverage in the car park environment. Then, the GN6ASL packages get converted vice versa into the car park protocol, which is then sent over Ethernet using TCP/IPv4 to the server, where the data processing and function execution is performed. The communication within the car park instances (gateway, server, car park sensors) can be done using a fixed IP address, as the participating instances are non-switched and do not change. The communication within the vehicles is also done using Ethernet. The vehicle ITS-S Gateway receives GN6ASL packages and converts them to the specific protocol using the proxy service. Afterwards, the protocol is sent over Ethernet to the vehicle electronic control unit (ECU), where the car park protocol and further information of on board sensors are merged and processed by vehicle functions. With the use of GeoNetworking features, an Ad-hoc network can be set up.

C. Vehicle Processing Chain

The complete processing chain inside the vehicle, beginning by the environment cognition up to the execution of driving actions, can be reflected in the SA model presented by Endsley

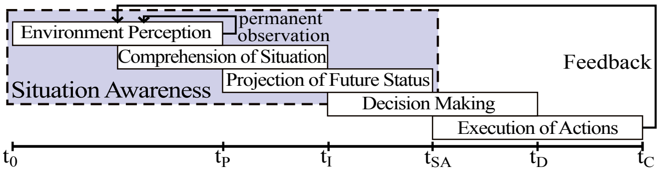


Fig. 3. Processing chain from situation awareness and decision making to the execution of driving events, comprising to [5].

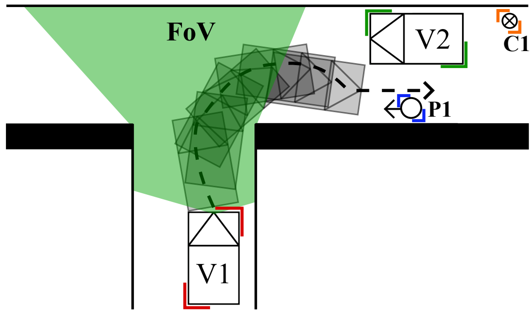


Fig. 4. Illustration of the scalable scenario including colour coding for participating objects and instances (cf. Fig 1). The ego-vehicle (V1) plans to turn right, while a third party vehicle (V2) moves to the same direction the ego-vehicle plans to move. In the same direction, a pedestrian (P1) is moving up to the T-crossing and a camera (C1) of the multi-storey car park is located to monitor the intersection. The initial FoV of V1 is illustrated as green plot.

[5]. The first steps focus on the situation interpretation. Then, specific functions take a decision and actuators perform driving actions. Every step has a specific runtime, which can be reflected in timestamps stepping out of a defined point of time. The execution of driving manoeuvres gives feedback to the environment and may influence actions of dynamic objects. Subsequently, the situation has to be under permanent observation. In the following, we describe the processing chain inside the vehicle in detail and give definitions for the runtime.

1) Situation Awareness: Following the model of Endsley [5], SA is divided into three steps. Step 1 includes the perception of the environment and the current situation. This step is reliant on the use case and may include various steps of procedure like processing sensor data, sensor frame rate, time for receiving shared environment information, time for data fusion, time for tracking and classifying objects and is fulfilled at timestamp t_P. Step 2 is the comprehension of the current situation, where the cognition of intention is performed. In this step, information about dynamic objects and elements, the road infrastructure, traffic rules and the self-state are linked and is presented as timestamp t_I. Afterwards, the prediction of future movement of the participating objects is processed to get a sufficient situation interpretation. In our scope, the situation awareness is fulfilled at timestamp t_{SA}. This includes the time for sharing the information with involved modules like ECUs.

2) Decision Making Process: After getting awareness of the current situation, a decision for future driving actions has to be made. The decision making process is normally processed on another module, component or ECU. For performing the decision making process, the whole information of situation awareness must be available. The runtime for decision making depends on the vehicle function and may vary for different applications. The decision making process is completely performed at timestamp t_D, when the decision is fully communicated to modules which perform driving actions.

3) Execution of Actions: In our case, a driving action can be defined as the performance of one or several combined actuators to reach a particular goal. This goal may be a specific point or stopping line, a target speed, a specific angle of the vehicle, a defined state of the vehicle, etc. The process of executing the driving action is starting, when the order for the manoeuvre completely arrived at one or more concerning vehicle modules. When the given target state of vehicle has been achieved, the process for executing the driving action is completed and is defined as timestamp t_C.

D. Scenario Modelling

In the scope of this paper, we are focusing on a scalable, closed environment car park scenario, where objects out of FoV use cases are usual events. We are assuming an ego-vehicle planning to make a right-turn action. The FoV is restricted due to walls on each side, as Fig. 4 illustrates. The vehicle on board sensor is a Lidar Infra-red Laser with a horizontal FoV of 145 degrees and a range of 150 meters. Beside of the FoV restriction, the width of the car park spaces is limited, so vehicles may intersect trajectories of foreign vehicles at a certain point of time, especially in turning manoeuvres vehicles have to step out of line.

1) Mixed traffic Scenario: The first experimental setup is a mixed traffic scenario, where information of the environment can only be interpreted by the ego-vehicles (V1) on-board sensors. The third party vehicle (V2) is seen as a classic human steered car with no possibility for V2X communication. Vehicle V1 is presented as an ADTF instance. Vehicle V2 is an VTD controlled object, as there is no need for a detailed function simulation. In this scenario, no pedestrian and no car park features are implemented. The ego-vehicle must be able to perform automated driving only with information generated by itself. As Fig. 4 shows, V2 is not in the FoV of V1 in the initial scene. This use case may take place after market launch of the first fully automated vehicles and in car park environments, where mixed traffic is allowed.

2) Car-to-Car Communication Scenario: The next scale of the scenario presents a use case where both vehicles communicate via Car-to-Car (C2C) technology and exchange their positions. Similar to Scenario 1, only V1 is set up as an ADTF instance. Vehicle V2 has the possibility to share its position. For simplicity reasons for the simulation setup, the position is shared using the VTD data bus. To take C2C issues into account, in this scenario empirical values are assumed. If latency for the C2C communication is needed, a framework for C2C communication can be integrated easily. The scope of this scenario is, how situation interpretation and decision making of V1 is influenced, when it has knowledge about V2 at a dedicated point of time, where V2 is not within the FoV of the ego-vehicle.

3) Collective Perception Scenario: Scenario 3 shows a use case, where the car park is accessible for pedestrians. Both vehicles, V1 and V2 are presented as separate ADTF instances. A pedestrian P1 (cf. Fig. 4, highlighted by blue corners), simulated by the *MomenTUMv2* framework, intersects a planned trajectory of V1. Pedestrian P1 is not within the FoV of V1, but can be recognized by the sensors of V2. After detecting P1, vehicle V2 sends information over C2C communication to V1. Following, V1 has environment information to a point of time, where the view of the on-board sensors is limited. In case of this scenario, the perception process (cf. Fig. 3) of both vehicles have to be conducted and evaluated. As a first step, latency values for C2C communication can be assumed. To get a more complex and detailed evaluation, the framework must be extended by tools for C2C simulation (i.e. *Veins* and *Artery*).

4) V2X Car Park Scenario: The last scale of the scenario presents a use case, where the decision making process can be done by an external server of the multi-storey car park. This use case may occur in multi-storey car parks which are limited to fully automated vehicles. In this scenario, the vehicles receive environment information from car park entities or only perform actions which are ordered by the car park logic. Following, the perception process, situation interpretation and the decision making can be done by the car park entities. Depending on the study, the vehicles can be presented as ADTF instances or objects in VTD. The initial scenario is extended by the car park implementation (cf. Fig. 1, orange highlighting), which includes tracking cameras, communication components and the server. In this setup, scenarios with situation interpretation and decision making on the car park server can be evaluated.

E. Metrics for Evaluation

When simulating and evaluating the described scenarios, we are focusing on latency of the involved steps of the processing chain. Thus, we are evaluating, how different points of time of information availability have impact on the situation interpretation and decision making process. Additionally, we analyse when latency in environment information exchange lead from comfortable driving actions to more immoderate actions like emergency breaking or deadlocks. Hence, the loss of planning options for future driving actions play a decisive role.

IV. Conclusion and Future Work

This contribution suggests a simulation setup to evaluate IVC concerning situation interpretation and decision making for fully automated vehicles. We introduced closed environment car park scenarios in order to investigate FoV use cases. Dependent on the scenario, participating vehicles have V2X technology implemented or have no possibility for V2X communication. The decision making can be done by the vehicles and the car park implementation. Future work includes the simulation and evaluation of each scenario and their modifications. Furthermore, we are planing to extend the simulation setup with frameworks for C2C and V2X communication (e.g. *Veins*). This enables a more detailed research about latency in the whole perception, interpretation, decision making and information sharing process.

References

[1] *Scenarios for Autonomous Vehicles- Opportunities and Risks for Transport Companies*, Position Paper, November 2015, Verband Deutscher Vehrkersunternehmen e.V. (VDV), Germany, [Online]. Available: https://www.vdv.de/position-autonomous-vehicles-vdv-engl.pdf. [Accessed: 01-03-2017]

[2] S. A. Beiker, *Einführungsszenarien für höhergradig automatisierte Straßenfahrzeuge*, In: M. Maurer, J.C. Gerdes, B. Lenz and H. Winner, (eds.) Autonomes Fahren: Technische, rechtliche und gesellschaftliche Aspekte. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015, pp. 197-217.

[3] S. Scherer, A. Dettmann, F. Hartwich, T. Pech, A. C. Bullinger and G. Wanielik, *How the driver wants to be driven - Modelling driving styles in highly automated driving*, In: 7. Tagung Fahrerassistenzsysteme, 2015.

[4] F. Hartwich, M. Beggiato, A. Dettmann, J. F. Krems, *Drive Me Comfortable. Customized Automated Driving Styles for Younger and Older Drivers*, In: 8. VDI-Tagung Der Fahrer im 21. Jahrhundert, VDI-Verlag, 2015, pp. 271-283.

[5] M. R. Endsley, *Toward a Theory of Situation Awareness in Dynamic Systems*, In: Human Factors: The Journal of the Human Factors and Ergonomics Society, 1995, vol. 37, pp. 32-64(33).

[6] M. W. B. Ghezala, A. Bouzeghoub, and C. Leroux, *RSAW: A situation awareness system for autonomous robots*, in ICARCV, 2014, pp. 450-455.

[7] J. A. Adams, *Unmanned vehicle situation awareness: A path forward*, In: Human systems integration symposium, 2007, pp. 31-89.

[8] Z. Lu, X. Coster, J. de Winter, *How much time do drivers need to obtain situation awareness ? A laboratory-based study of automated driving*, In: Applied Ergonomics, Volume 60,2017, pp. 293-304.

[9] D. Miller, A. Sun, W. Ju, *Situation awareness with different levels of automation*, In: 2014 IEEE International Conference on Systems, Man, and Cybernetics, San Diego, CA, USA 2014, pp. 688-693.

[10] Forschungsinitiative Ko-FAS Verbundprojekt Ko PER, *Ko-PER spezifische Nachrichteninhalte ASN1 Spezifikation (öffentliche Fassung) MS 2.720-2*. In: Ko-FAS Deliverables, 2012.

[11] L. Hobert *Autonet 2030 - Specifications for the enhancement to existing LDM and cooperative communication protocol standards*, In: Autonet 2030 Deliverable 3.2., 2015.

[12] A. Rauch, F. Klanner, and K. Dietmayer. *Analysis of V2X communication parameters for the development of a fusion architecture for cooperative perception systems*, In: Intelligent Vehicles Symposium (IV). June 2011, pp. 685-690.

[13] A. Rauch, F. Klanner, R. Rasshofer, and K. Dietmayer. *Car2X-Based Perception in a High-Level Fusion Architecture for Cooperative Perception Systems*, In: Intelligent Vehicles Symposium (IV). June 2012, pp. 270-275.

[14] H. J. Günther, B. Mennenga, O. Trauer, R. Riebl and L. Wolf, *Realizing collective perception in a vehicle*, In: 2016 IEEE Vehicular Networking Conference (VNC), Columbus, OH, 2016, pp. 1-8.

[15] H. J. Günther, R. Riebl, L. Wolf and C. Facchi, *Collective perception and decentralized congestion control in vehicular ad-hoc networks*, In: 2016 IEEE Vehicular Networking Conference (VNC), Columbus, OH, 2016, pp. 1-8.

[16] R. Riebl, H.-J. Günther, C. Facchi and L. Wolf. *Artery- Extendig Veins for VANET applications*. In: Models and Technologies for Intelligent Transportation Systems (MT-ITS), 2015.

[17] M. Behrisch, L. Bieker, J. Erdmann and D. Krajzewicz, *SUMO - Simulation of Urban MObility: An Overview*, In: SIMUL 2011, The Third International Conference on Advances in System Simulation.

[18] C. Sommer, R. German, and F. Dressler, *Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis*, In: IEEE Transactions on Mobile Computing 10.1, 2011, pp. 3-15.

[19] P. M. Kielar, D. H. Biedermann, A. Borrmann, *MomenTUMv2: A Modular, Extensible, and Generic Agent-Based Pedestrian Behavior Simulation Framework*, In: TUM-I1643, 2016, Technische Universitt Mnchen.

Protocol Modeling Accuracy in VANET Simulators

Sebastian Bittl*, Karsten Roscher†
*HU Berlin, sebastian.bittl@mytum.de, †Fraunhofer ESK, karsten.roscher@esk.fraunhofer.de

***Abstract*—Vehicular ad hoc networks are about to enter the mass market in upcoming years. High effort for real world field tests leads to high dependency of development and evaluation of such networks on simulations. We compare supported features of common simulation frameworks with current standards and study the performance impact of incomplete standard conformance. We find that a lack of support for data encoding schemes and security functionality may massively affect simulation results. Our findings apply to many well known simulation frameworks. Proposals to overcome identified weaknesses are provided.**

***Index Terms*—Simulation, Evaluation, Performance, VANET;**

I. Introduction

Vehicular ad hoc networks (VANETs) are of high interest to enable future cooperative advanced driver assistance systems (ADAS). Such ADAS are typically intended to serve as safety critical collision avoidance systems, which are expected to significantly increase safety of driving [1]. Standardization of VANET technologies is performed within European ETSI Intelligent Transport Systems (ITS) and US Wireless Access in Vehicular Environments (WAVE) frameworks [1], [2].

Evaluation of VANET protocols and applications is mostly based on simulations. This is caused by the high effort to perform real life tests in a reproducible way with many participating vehicles. Even large scale field tests, like DriveC2X or SimTD in Europe, achieved only a volume of 18 to 30 VANET ready vehicles per test site [3]. Moreover, standards have changed significantly since such tests were conducted and repetition for each standard version seems infeasible. Thus, most published results about VANETs depend on simulations.

To evaluate how accurate available simulation frameworks resemble standardized VANET protocols, a study on supported features and the impact of missing features on simulation behavior is provided. We find that all open source and also some commercial frameworks lack support of features showing significant impact on VANET performance. Suggestions on how missing features can be provided with limited effort are given.

The remainder of this work is outlined as follows. Related work is looked at in Section II. State of the art VANET simulation is reviewed in Section III. An evaluation of the impact of identified shortcomings on simulation behavior is provided in Section IV. Finally, Section VI gives a conclusion alongside with possible topics of future work.

II. Related Work

VANET simulation frameworks are studied in [4], [5]. However, many regarded frameworks are no longer maintained, and new ones have emerged in the meantime. Moreover, many changes have been applied to VANET standards. [1] gives an overview of three frameworks (Veins, iTETRIS, VSimRTI), but no comparison is provided. Thus, we provide an up to date study of available simulation frameworks in this work.

Prior work has identified numerous requirements on VANET simulation frameworks. These include among others dedicated channel models for vehicular environments, realistic driver models for node movement and inclusion of at least some hundred nodes per traffic scenario [4]–[7].

Different message formats are used within WAVE and ETSI ITS. Both systems use so called beacon messages for basic information exchange. Cooperative Awareness Messages (CAMs) are used in ETSI ITS, while WAVE utilizes Basic Safety Messages (BSMs). Moreover, ETSI ITS uses optionally sent Decentralized Environment Notification Messages (DENMs), while WAVE includes corresponding data on demand as an optional second part of the BSM [1]. Dedicated simulation frameworks have been developed for both ETSI ITS and WAVE. However, we show that simulation based comparison of both systems is hardly possible, due to various missing standardized features in simulators for both domains.

Many simulation frameworks compromise a combination of multiple dedicated simulators performing highly specialized tasks. Popular components are ns-3 or OMNET++ for network simulation (physical and MAC layer) [8], [9] and SUMO (Simulation of Urban Mobility) [10] for microscopic traffic flow simulation. More details are given in Section III.

III. Analysis of State of the Art

A VANET simulator needs to accurately model different sub-aspects relevant to a VANET. Individual movement of each vehicle and channel modeling are core points of concern in regard to the design of a VANET simulator [4]–[7]. To achieve reliable simulation results for VANETs, different methodologies have been proposed. Often a combination of dedicated and well known simulators for the individual sub-aspects is used, e.g., one for traffic flow, one for channel modeling and another one for communication protocol behavior.

Many simulation environments have been published focusing on different aspects of VANET communication. An overview is given in Table I. Earlier developed frameworks TraNS, GrooveNet, NCTUns (now: EstiNet) and MobiREAL, all studied in [4], [5], are not maintained any more. Hence, they are not regarded in the following. As given in Table I, VANETSim, VNS (Vehicular Network Simulator) and EstiNet do not aim to provide a full VANET standard compatible protocol stack. For example, EstiNet focuses on software defined networks (SDNs). Thus, these frameworks are not in the focus of our work. Moreover, VSimRTI [15] is not taken into regard any further, due to a lack of access to this commercial tool.

Comparison of the simulation environments from Table I shows that open source VANET implementations lack at least partly four main features from current standards. These are

1) security mechanisms of WAVE and ETSI ITS [19], [20],

TABLE I
COMPARISON OF FEATURES OF DIFFERENT SIMULATION FRAMEWORKS FOR VANETS.

name	open source	focus	missing standardized features	traffic flow simulator	network simulator
Veins [7]	yes	WAVE	security, data encoding, time/position coupling	SUMO	OMNET++
Artery [11]	yes	ETSI ITS	outdated security, time/position coupling, ITS-G5 (802.11p used instead), DCC	SUMO	OMNET++
iTETRIS [12]	yes	ETSI ITS	security, ASN.1, time/position coupling	SUMO	ns-3
VANETSim [13]	yes	security	no standard compliance	own	own
VNS [14]	yes	traffic flow	no standard compliance	own	ns-3, OMNET++
VSimRTI [15]	no	ETSI ITS	DCC	SUMO, VISSIM [16]	ns-3, OMNET++
ezCar2X [17]	no	ETSI ITS	-	SUMO	ns-3
EstiNet [18]	no	SDN	security, all above network layer	own	own

2) data encoding by ASN.1, e.g., for CAM, DENM as well as WAVE network layer [1], [21], [22],
3) coupled updating of time and position information within the protocol stack [21]–[23],
4) distributed congestion control (DCC) [21], [24].

Such aspects significantly influence the size of messages transmitted by each VANET node. Security overhead in messages was shown to have a major influence on experienced VANET performance [25], [26]. This lead to the development of many different approaches to limit this impact [25]–[28]. Influence of different data encoding schemes for ETSI ITS data sets is also significant in regard to achieved message size [29].

The lack of support for DCC within Artery is caused by using an 802.11p model for the physical and MAC layer. ETSI ITS's DCC requires channel state information from the ITS-G5 MAC layer, which is not provided by an 802.11p MAC layer.

The size of messages transmitted by VANET participants has a major impact on experienced channel load. This holds especially for VANETs with fixed message transmission rate, e.g., WAVE [1]. Within ETSI ITS transmission ratios can be reduced from a maximum of 10 Hz down to 1 Hz, due to the monitored channel load [21]. This can limit the channel load increase in some scenarios. However, it decreases the update rate of receivers, which limits data quality for corresponding applications. Furthermore, VANETs suffer from decreasing communication range in case of increasing channel load. This is mainly caused by the hidden station problem causing an increasing amount of collisions on the wireless channel alongside with an increase in channel load [30].

Limited reliable communication range is a core point of concern for the design of cooperative ADAS. Reduced communication range leads to lower reaction times for ADAS. Thus, much effort is taken to keep channel load low [25], [26], [28], [31], [32] and the achieved communication range is part of many VANET performance metrics (see e.g., [28]).

One can state that increased message size leads to degradation of experienced data quality in VANETs. However, the impact on an application's performance depends on its dedicated requirements and the considered traffic scenario.

Another impact of missing security mechanisms on communication behavior is caused by the absence of cryptographic packet loss, which is caused by the inability of receivers to verify packets' signatures, due to sporadic inclusion of public keys [19], [28]. Thus, absence of security mechanisms in simulations increases the packet delivery rate to applications in comparison to implementations realizing security mechanisms.

Moreover, most simulation frameworks do not use coupled updates for time and position data sets at each node (see Table I). Instead, time is directly derived from simulation time. In contrast, position updates are performed whenever synchronization to the traffic flow simulator happens, i.e., nodes are static between such updates. Typically, intervals between position updates are much longer then time stamp increments. Thus, a deviation of an node's real trajectory from the one modeled by emitted messages occurs. Hence, data accuracy at receivers is lowered affecting all protocol layers using such data. Thus, robustness of applications relying on such data can be expected to suffer. E.g., this affects ADAS at the application level, but also position based routing at the network layer.

Standards specify to only update time and position data sets together, e.g., every time a new GPS fix is available [21]–[23]. The impact of missing coupling of updates for both data sets is illustrated in Figure 1 for a constant velocity. s gives the length of the traveled path and t represents the corresponding traveling time. Trajectory points (i.e., a time stamp together with a position) obtained from using current time, but only sporadically updated positioning data, are always found in the marked area of Figure 1. However, correct values are only the ones on the line giving the upper limit of that area. The dedicated deviation depends on the elapsed time since the last position update. Rapid position updates can limit the impact of the found design weakness, but cannot overcome it completely.

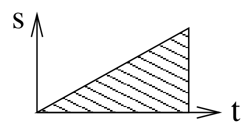


Fig. 1. Trajectory deformation from missing time and position coupling.

DCC mechanisms have been shown to influence CAM emission frequencies and channel access behavior [33], [34]. Hence, a lack of support for DCC will yield a VANET with a channel load significantly differing from an approach using DCC mechanisms. This holds especially for high density traffic scenarios, which yield high channel load being limited by DCC.

A. Impact on Evaluation of Cross Layer Functionality

Missing supported features on various protocol layers make it impossible to evaluate their corresponding impact on overall system performance. Examples of recently found issues of ETSI ITS, which cannot be identified by incomplete evaluation environments like Artery or iTETRIS, are given by,

- absolute location dependent inability to send the largest part of regular messages, due to maximum messages size violations (from DCC) on the MAC layer [35]

- impossible encrypted multi-hop communication, due to encryption of data sets needed for forwarding [36].

Moreover, usage of incomplete evaluation environments may lead to proposals of inappropriate approaches, e.g., duplicated data on various protocol layers. As an example, adding the so called ITS-Application Identifier (ITS_AID) to the GeoNetworking header is suggested in [37]. However, this data set is already present in the security envelope and handing over this data to the network layer is part of the standardized interface between both parts of the protocol stack [19]. Thus, including the ITS_AID into network layer meta data is pure overhead.

B. Impact on Simulation Performance

Some of the regarded frameworks claim very high simulation performance, e.g., [13]. However, this is at least partially caused by missing features. E.g., a lack of security features implies skipping of many validity checks, like signature verifications. Thus, a performance comparison between different frameworks has to be regarded as being unfair. Instead, performance of frameworks providing an equal feature set should be compared.

Adding missing features will affect the performance of simulation frameworks, especially regarding data encoding and cryptographic operations [29]. To speed up simulations without attack scenarios, generation and verification of signatures can be skipped, i.e., corresponding data is replaced by random values of the same size. The purpose of signatures is not required in that case, i.e., verification always succeeds anyway.

However, such an approach cannot be used for data encoding in general. The length of encoded data sets depends on the dedicated content which gets encoded, e.g., for ASN.1 (U)PER and binary encoding of the security envelope.

IV. Evaluation

To evaluate the impact of the found incomplete protocol implementations on VANET simulator behavior, we study the performance criteria of beacon message (CAM / BSM) size at the MAC layer. The tested CAMs and BSMs are created by using only mandatory fields. The security entity uses certificates corresponding to the example given in [19].

Obtained results for beacon messages sizes, as used within different simulation frameworks, are given in Table II. In all cases beacons not holding a pseudonym certificate are considered. Including a pseudonym certificate increases the standardized size by 115 bytes [19], but only for Artery and ezCar2X an increase in the simulated messages' size follows. In contrast, iTETRIS is found to use a constant, content independent message size. Hence, this framework cannot be used to evaluate on-demand data dissemination schemes, e.g., sporadic sending of the low frequency container in CAMs or pseudonym certificates in the security envelope of messages.

TABLE II
BEACON MESSAGE SIZES AT MAC LAYER IN BYTES.

	std.	Veins	Artery	iTETRIS	ezCar2X
BSM	215	70	-	-	-
CAM	217	-	221	300	217

To resemble VANET performance in an accurate and reliable way, the difference between the standardized message size (first column) and the obtained messages sizes form the different simulator implementations (other columns in Table II) should be zero. To obtain standard conformant results, in detail inspection of corresponding standards was used. For dedicated sizes of content added at different protocol layers of a CAM see, e.g., [19], [23], [35]. Corresponding data for BSMs can be obtained from [20], [38]–[40]. The data size results for the simulation frameworks were obtained by adding extra code to their MAC layer implementations, which measures the actual data size of sent messages at this layer.

Figures for BSM in Table II show a significant gap between the expected, i.e., standardized, message size and the one being used by Veins. In details inspection of the Veins's source code shows that this framework does not use any VANET specific data representation. Instead, the generic message data representation format of OMNET++ is used. Moreover, no security envelope is used, which accounts for the biggest share of missing message size (96 bytes). The obtained message size deviation can be expected to lower the channel load to about one third in comparison to the one experienced by using the standardized approach. Clearly, an impact on evaluation results for applications can be expected from such a difference.

The small difference between the current standardized version and the result obtained from Artery is due to the usage of an outdated security standard [41] in Artery. The current size of the security envelope is slightly reduced in comparison to this older standard version [19].

These findings do not imply that results obtained from well known simulation frameworks are incorrect in general. However, their portability on performance of real world implementations is called into question, as simulation frameworks do not represent standardized real world systems well.

V. Overcoming the Obtained Issues

ETSI ITS uses unaligned packet encoding rules (UPER) for ASN.1 [1]. Unfortunately, common open source ASN.1 tools do not support this kind of encoding [42]. However, project i-Game provides such an implementation dedicated to features required within ETSI ITS [43]. Other ASN.1 variants used in protocol stack standards are well supported by open source software [42]. Thus, there is no real burden for adding ASN.1 support to common VANET simulation frameworks.

Coupled handling of time and position information is illustrated by the ETSI ITS implementation of project i-Game [43].

Basic security mechanisms required by ETSI ITS and WAVE are provided by well known security frameworks, e.g., Crypto++ [44], which is used by ezCar2X and Artery. Moreover, parsing of the binary encoded security envelope from ETSI ITS is implemented within these two simulators, too.

VI. Conclusion

VANETs are considered as an important step towards future cooperative ADAS for increased safety of driving. High effort of practical field tests lead to a high dependency of development of protocols and applications on simulation based evaluation.

We compare the features provided by popular VANET simulation environments regarding their resembling of standardized protocol stacks. We find that significant deviations of the implemented communication behavior from standards exists. Our analysis shows that these deviations mostly lead to less challenging communication conditions in comparison to a system using a standardized protocol stack. Thus, usage of such simulation environments may yield too optimistic results for evaluated VANET mechanisms. Hence, we propose to extend the frameworks by the missing standardized components, especially ASN.1 data encoding and security mechanisms.

Future work can compare evaluation results for applications used for both ETSI ITS and WAVE, to show how different system design choices affect application performance.

REFERENCES

[1] C. Campolo, A. Molinaro, and R. Scopigno, Eds., *Vehicular ad hoc Networks - Standards, Solutions, and Research*. Springer, Dec. 2015.

[2] J. Harding et. al., "Vehicle-to-Vehicle Communications: Readiness of V2V Technology for Application," Washington, DC: National Highway Traffic Safety Administration, Tech. Rep. DOT HS 812 014, Aug. 2014.

[3] "test sites - Drive C2X," online: http://www.drive-c2x.eu/test-sites, 2014, retrieved: 04.2016.

[4] F. J. Martinez, C. K. Toh, J.-C. Cano, C. T. Calafate, and P. Manzoni, "A Survey and Comparative Study of Simulators for Vehicular ad hoc Networks (VANETs)," *Wireless Communications & Mobile Computing*, vol. 11, no. 7, pp. 813–828, July 2011.

[5] F. Dressler, C. Sommer, D. Eckhoff, and O. K. Tonguz, "Toward Realistic Simulation of Intervehicle Communication: Model, Techniques and Pitfalls," *IEEE Vehicular Technology Magazine*, vol. 6, no. 3, pp. 43–51, Sept. 2011.

[6] M. Boban, "Realistic and Efficient Channel Modeling for Vehicular Networks," PhD thesis, Dept. of Electrical and Computer Engineering, Carnegie Mellon University, Dec. 2012.

[7] 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.

[8] G. F. Riley and T. R. Henderson, "The ns-3 Network Simulator," in *Modeling and Tools for Network Simulation*, K. Wehrle, M. Günes, and J. Gross, Eds. Springer Berlin Heidelberg, 2010, pp. 15–34.

[9] A. Varga, *Modeling and Tools for Network Simulation*. Springer, 2010, ch. OMNeT++, pp. 35–59.

[10] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "SUMO - Simulation of Urban MObility: An Overview," in *The Third International Conference on Advances in System Simulation*, Oct. 2011, pp. 63–68.

[11] R. Riebl, H. J. Gunther, C. Facchi, and L. Wolf, "Artery: Extending Veins for VANET Applications," in *4th International Conference on Models and Technologies for Intelligent Transport Systems*, June 2015, pp. 450–456.

[12] M. Rondinone, et. al., "iTETRIS: A modular simulation platform for the large scale evaluation of cooperative ITS applications," *Simulation Modelling Practice and Theory*, vol. 34, pp. 99–125, May 2013.

[13] A. Tomandl, D. Herrmann, K.-P. Fuchs, H. Federrath, and F. Scheuer, "VANETsim: An open source simulator for security and privacy concepts in VANETs," in *International Conference on High Performance Computing Simulation*, July 2014, pp. 543 – 550.

[14] R. Fernandes, F. Vieira, and M. Ferreira, "VNS: An Integrated Framework for Vehicular Networks Simulation," in *IEEE Vehicular Networking Conference*, Nov. 2012, pp. 195 – 202.

[15] B. Schuenemann, "V2X Simulation Runtime Infrastructure VSimRTI: An Assessment Tool to Design Smart Traffic Management Systems," *Computer Networks*, vol. 55, no. 14, pp. 3189–3198, Oct. 2011.

[16] M. Fellendorf and P. Vortisch, "Microscopic traffic flow simulator VISSIM," in *Fundamentals of Traffic Simulation*. Springer New York, 2010, ch. 2, pp. 63–93.

[17] K. Roscher, S. Bittl, A. A. Gonzalez, M. Myrtus, and J. Jiru, "ezCar2X: Rapid-Prototyping of Communication Technologies and Cooperative ITS Applications on Real Targets and Inside Simulation Environments," in *11th Conference Wireless Communication and Information*, Oct. 2014, pp. 51 – 62.

[18] S.-Y. Wang, C.-L. Chou, and C.-M. Yang, "EstiNet Openflow Network Simulator and Emulator," *IEEE Communications Magazine*, vol. 51, no. 9, pp. 110 – 117, Sept. 2013.

[19] *Intelligent Transport Systems (ITS); Security; Security header and certificate formats*, ETSI TS 103 097, Rev. V1.2.1, 2015.

[20] *IEEE Standard for Wireless Access in Vehicular Environments - Security Services for Applications and Management Messages*, Intelligent Transportation Systems Commitee of the IEEE Vehicular Technology Society Std. P1609.2, Rev. 2013, Apr. 2013.

[21] *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service*, ETSI ES 302 637-2, Rev. V1.3.2, Nov. 2014.

[22] *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service*, ETSI ES 302 637-3, Rev. V1.2.1, Sept. 2014.

[23] *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*, ETSI ES 302 636-4-1, Rev. V1.2.1, July 2014.

[24] *Intelligent Transport Systems (ITS); Decentralized Congestion Control Mechanisms for Intelligent Transport Systems operating in the 5 GHz range; Access layer Part*, ETSI TS 102 687, Rev. V1.1.1, July 2011.

[25] F. Kargl, E. Schoch, B. Wiedersheim, and T. Leinmüller, "Secure and Efficient Beaconing for Vehicular Networks," in *Fifth ACM international workshop on VehiculAr Inter-NETworking*, 2008, pp. 82–83.

[26] E. Schoch and F. Kargl, "On the Efficientcy of Secure Beaconing in VANETs," in *Proceedings of the Third ACM Conference on Wireless Network Security*, Mar. 2010, pp. 111 – 116.

[27] G. Calandriello, P. Papadimitratos, J. P. Hubaux, and A. Lioy, "On the Performance of Secure Vehicular Communication Systems," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 6, pp. 898 – 912, Sept. 2011.

[28] M. Feiri, J. Petit, R. Schmidt, and F. Kargl, "The Impact of Security on Cooperative Awareness in VANET," in *IEEE Vehicular Networking Conference*, Dec. 2013, pp. 127 – 134.

[29] S. Bittl, A. A. Gonzalez, M. Spähn, and W. Heidrich, "Performance Comparison of Data Serialization Schemes for ETSI ITS Car-to-X Communication Systems," *International Journal On Advances in Telecommunications*, vol. 8, pp. 48 – 58, June 2015.

[30] R. K. Schmidt, T. Köllmer, T. Leinmüller, and G. Schafer, "Degradation of Transmission Range in VANETs caused by Interference," *PIK - Praxis der Informationsverarbeitung und Kommunikation*, vol. 32, no. 4, pp. 224 – 234, Jan. 2010,

[31] M. Feiri, J. Petit, and F. Kargl, "Evaluation of Congestion-based Certificate Omission in VANETs," in *IEEE Vehicular Networking Conference*, Nov. 2012, pp. 101 – 108.

[32] B. Kloiber, T. Strang, F. de Ponte-Mueller, C. Rico Garcia, and M. Roeckl, "An Approach for Performance Analysis of ETSI ITS-G5A MAC for Safety Applications," in *The 10th International Conference on Intelligent Transport Systems Telecommunications*, Nov. 2010.

[33] G. Bansal, B. Cheng, A. Rostami, K. Sjoberg, J. B. Kenney, and M. Gruteser, "Comparing LIMERIC and DCC Approaches for VANET Channel Congestion Control," in *IEEE 6th International Symposium on Wireless Vehicular Communications*, Sept. 2014, pp. 1–7.

[34] A. Festag, "Cooperative Intelligent Transport Systems Standards in Europe," in *IEEE Communications Magazine*, vol. 52, no. 12, Dec. 2014, pp. 166–172.

[35] S. Bittl, K. Roscher, and A. A. Gonzalez, "Security Overhead and its Impact in VANETs," in *8th IFIP Wireless Mobile Networking Conference*, Oct. 2015, pp. 192 – 199.

[36] S. Bittl, "Towards Solutions for Current Security Related Issues in ETSI ITS," in *10th International Workshop Nets4Cars/Nets4Trains/Nets4Aircraft*, ser. LNCS, vol. 9669, May 2016, pp. 136 – 148.

[37] C. Büttner and S. A. Huss, "An Anonymous Geocast Scheme for ITS Applications," in *2nd International Conference on Information Systems Security and Privacy*, Feb. 2016, pp. 179–189.

[38] *IEEE Standard for Wireless Access in Vehicular Environments (WAVE) - Networking Services*, Intelligent Transportation Systems Commitee of the IEEE Vehicular Technology Society Std. P1609.3, Rev. 2010, Dec. 2010.

[39] D. Kelly, "DSRC Implementation Guide," SAE International, Tech. Rep., Feb. 2010.

[40] *IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Amendment 6: Wireless Access in Vehicular Environments*, IEEE Std., Rev. 802.11p-2010, July 2010, 802.11p-2010.

[41] *Intelligent Transport Systems (ITS); Security; Security header and certificate formats*, ETSI TS 103 097, Rev. V1.1.1, 2013.

[42] ITU, "ASN.1 Project Tools," online: http://www.itu.int/en/ITU-T/asn1/Pages/Tools.aspx, 2016, retrived: 01.2016.

[43] i-GAME GCDC, "GCDC - Rules & Technology," online: http://www.gcdc.net/teams/rules-and-technology, 2016, retrieved: 01.2016.

[44] W. Dai, "Crypto++ Library," online: http://www.cryptopp.com/, Nov. 2015, retrieved: 07.2016.

Proceedings of the 5th GI/ITG KuVS Fachgespräch Inter-Vehicle Communication (FG-IVC 2017)
Anatoli Djanatliev, Kai-Steffen Hielscher, Christoph Sommer, David Eckhoff, and Reinhard German (Eds.)

Towards the Evaluation of Three-Dimensional Scenarios in VANET Simulation

Alexander Brummer*, Anatoli Djanatliev*
* Computer Networks and Communication Systems, Dept. of Computer Science, University of Erlangen, Germany
{alexander.brummer,anatoli.djanatliev}@fau.de

***Abstract*—Simulation frameworks for the evaluation of VANET applications and protocols have recently reached a high level of realism. However, most of these simulators are based on a planar representation of the underlying road network and environment. As a consequence, a lot of scenarios as can be found in real-world environments cannot be simulated appropriately, e.g. in case of situations including bridges and inclined streets, communication with elevated infrastructure or road users of different heights. In order to get more reliable results for these kinds of scenarios, a three-dimensional simulation environment becomes necessary. In this paper, we first outline related work emphasizing the importance of a three-dimensional consideration. Afterwards, we discuss the requirements for enabling three-dimensional VANET simulation. This includes possible ways of adding elevation information to the traffic simulation as well as implications and necessary extensions for the network simulation such as three-dimensional antenna patterns. Furthermore, we present two selected scenarios, which require three-dimensional investigations for more realistic results.**

Figure 1. Example of a three-dimensional VANET scenario featuring a bridge across a highway.

I. INTRODUCTION

Vehicular communication is a promising technology which is based on the exchange of information between traffic users. With the help of Vehicular Ad-Hoc Networks (VANETs), it is possible to implement new technologies and applications that can lead to higher level of comfort and safety in modern traffic. For the investigation and evaluation of such applications and protocols simulations have become a key tool [1].

The degree of realism in vehicular networking simulation has constantly increased in recent years of research. On the one hand, a variety of models to properly represent the environment have been developed, such as the obstruction by buildings [2]. On the other hand, diverse technology-related effects have been taken into account in order to obtain realistic models for the different layers [3], [4]. Combining detailed network simulation with mobility simulation enables the realistic, bidirectionally influencing of both parts, which is an essential property of vehicular communications [5].

Despite these improvements, almost all simulation frameworks still neglect another important aspect: They are mostly based on a two-dimensional environment, considering the horizontal plane only. This simplification thereby ignores individual characteristics of various real-world scenarios. Obviously, the reason is that one won't find neither all communication partners nor obstacles at the same heights. This is caused by the fact that the geometry of VANETs directly depends on the environment of the specific traffic situation in question.

Examples for emphasizing this issue can be found in urban, suburban and also highway scenarios. One may simply think of a road crossing underneath a bridge (see Figure 1). In a planar approach, this constellation might seem like a usual intersection and simulating the scenario this way would lead to similar results. In fact, however, the different altitudes of communicating vehicles can have a big impact on the quality of communication, caused by possibly different signal paths, antenna gains or obstruction by the bridge itself.

As soon as the z-coordinate of participating vehicles might make the difference, a three-dimensional investigation and thus simulation can deliver important insights. Therefore, prominent 3D-scenarios do not need to be based on bridges only, but can also be justified by antennas being mounted on vehicles and/or infrastructure of differing heights, multi-storey parking garages, hilly terrain and so forth.

In this paper, we want to outline the importance of three-dimensional VANET simulation and how this can be achieved. Therefore, we first summarize related work concerning three-dimensional considerations in the context of vehicular networks. Afterwards, the necessary steps to realize a preferably realistic simulation of three-dimensional scenarios are described and possible issues are identified. We further present two selected scenarios of interest in more detail to show the necessity and the challenges of 3D considerations. Finally, we summarize our insights towards three-dimensional vehicular network simulation.

II. Related Work

Three-dimensionality in VANETs has already been researched in different contexts. Several studies investigate the influence of the third dimension on the performance of routing protocols. In [6], a Three-Dimensional Scenario Oriented Routing (TDR) protocol is presented. The authors argue that common greedy forwarding mechanisms assuming a planar environment lead to a higher number of hops and a lower delivery ratio when vehicles on different road layers try to communicate. Taking the z-coordinate into account, the hop count could be reduced by 23 % and the delivery ratio increased by 40 %. The influence of different road levels on the forwarding method has also been discussed in [7], showing a significant dependence of delay and delivery ratio on the overpass height.

Besides these consequences for higher layer protocols, the influence of 3D environments on lower layers has been noticed as well. Boban et al. [8] investigated the signal attenuation caused by other vehicles. As most of the common cars have similar heights, the Line of Sight (LOS) is often obstructed by other vehicles between sender and receiver, causing significant packet loss. In [9], the authors present a Tall Vehicle Relaying (TVR) method, which makes use of the better propagation properties of taller vehicles by intentionally choosing them as next hops.

Another well-known aspect is the impact of antenna patterns. In general, antennas can show strong directedness, causing critical additional gains or attenuations depending on the relative positions of sender and receiver. In [10], [11], the authors show that the type as well as the mounting point of the antennas in use can lead to significantly varying antenna patterns. For a detailed analysis, ray-tracing simulations have been performed. Although being very accurate and aware of 3D environments, this type of investigation is a very time-consuming and computationally expensive task. Moreover, ray-tracing requires a highly detailed and scenario-specific model of the environment. In this context, the consideration of antenna patterns in packet-based simulation has been researched in [12]. It could be shown that the strong directedness in VANETs as well as characteristic antenna patterns can have negative influence on safety applications such as intersection collision avoidance. However, the support for antenna patterns was still restricted to two-dimensional scenarios only.

III. Requirements for 3D VANET Simulation

As the small introductory example as well as the related work indicate, three-dimensional simulation of vehicular networks is necessary in order to obtain more realistic results. The steps for this extension are outlined in the following, based on the open-source framework Veins[1]. Veins combines the network simulator OMNeT++[2] with the traffic simulation tool SUMO[3] by bidirectionally coupling both [5]. Therefore, certain adjustments in both parts have to be made.

[1]http://veins.car2x.org/
[2]https://omnetpp.org/
[3]http://sumo.dlr.de/

A. Supporting Three-Dimensional Road Networks

Mobility simulation can be seen as the foundation of VANET simulation as the geometry of the resulting network is determined by the specific traffic situation in question. As a consequence, an important requirement are three-dimensional road networks, so that their shape of the investigated scenario can be modeled appropriately.

In fact, SUMO's network files already support the usage of the z-coordinate in order to state a node's altitude. Here, the challenge is how to get reasonable elevation data and how to import it.

1) Building Networks Manually: For smaller, artificial scenarios, the most trivial option is to build up a SUMO network by hand, e.g. for modeling a parking garage as described in Section IV-A. To do so, it is possible to define a set of nodes, which are connected by a set of edges, both annotated in XML. Edges themselves can be further modeled by making use of a shape attribute. Stating 3D coordinates for all of these nodes creates a three-dimensional road network, which can then be converted to a proper SUMO network file with help of the tool NETCONVERT.

2) Importing measured elevation information: Obviously, creating such artificial scenarios is not sufficient. For realistic VANET simulation, real road networks have to be loaded. In SUMO, it is possible to import maps from the OpenStreetMap project[4]. Unfortunately, these maps almost never include altitude data, thus importing such a map only is not enough if one aims for three-dimensional simulation.

However, it is possible to add elevation data using so called Digital Elevation Models (DEMs). These are datasets containing topographic data, i.e. elevation information for a certain region of interest. A relatively detailed, nearly global DEM was generated by the Shuttle Radar Topography Mission (SRTM) [16]. Using interferometric radar technology, a resolution of about 30 m could be achieved. There exists a plugin for OpenStreetMap which takes a map as input and adds altitude information from the SRTM datasets. The resulting map can then be converted to a SUMO network file representing the three-dimensional road network. It is further possible to directly import heightmaps into an existing SUMO network using the NETCONVERT tool. In this case, all kinds of sources such as elevation models provided by geographic or official institutes can be used given that format and coordinate system are supported.

Another source for elevation data is the Google Maps Elevation API[5]. Sending a request to this web service by Google delivers the altitude information for this point on Earth. This can be done for a single location, a set of arbitrary points or a set of places along a certain path. In [17], the authors describe how this source can be used to add elevation information to an existing, two-dimensional SUMO network file for the purpose of electric vehicle simulation.

[4]http://www.openstreetmap.org
[5]https://developers.google.com/maps/documentation/elevation/intro

An issue that comes with the usage of DEMs is the fact that they represent the topmost surface (which reflected the radar beam). Therefore, problems arise in case of bridges, tunnels and similar sceneries. Here, the lower layers won't be described correctly as there is simply no data present in the DEM. Such points have to be adjusted manually.

B. Extending the Network Simulation

As soon as three-dimensional road data has been realised, also the network simulation has to be extended accordingly in order to be able to make use of the new altitude information.

1) Supporting the z-coordinate: First of all, the simulation environment simply has to incorporate the third dimension in all calculations concerning the physical layer. In OMNeT++, the MiXiM framework [4] is widely-used for the simulation of the lower layers in wireless networks. For realistic VANET simulation, appropriate models for the IEEE 802.11p standard have been added [3].

In principle, MiXiM already supports the z-coordinate, however, almost no models utilize it. Therefore, the concerning implementations need to be extended by a three-dimensional point of view to ensure the appropriate usage of elevation data.

Furthermore, the Traffic Control Interface (TraCI) has to be adjusted similarly. It is responsible for the exchange of information between mobility and network simulation and has to function properly in order to keep track of the added height information.

2) Additional attenuation models: With the introduction of three-dimensional environments, new obstacles causing further signal attenuation have to be considered. On the one hand, a possibly vertical signal path in the context of bridges or parking garages (see Section IV-A) might be heavily attenuated by the floors. The influence of common buildings and walls has already been investigated by Sommer et al. [2]. Their insights might potentially be transferred to the case of "horizontal walls".

On the other hand, the influence of other vehicles can play a critical role as already mentioned in Section II. In [8], the authors also describe a model for the attenuation induced by other road users. Based on that, a new model for the Veins framework could be conceivable.

3) Three-dimensional antenna patterns: The possibility to assign general antenna patterns to vehicles in the first place has been described in [12]. In this modular approach, it is possible to add new antenna types by simply adding antenna classes representing the desired functionality. In the process, a class representing sampled, horizontal antenna patterns has already been implemented. The support for 3D antenna models can thus be ensured by adding another class that deals with horizontally *and vertically* sampled antenna patterns. For the implementation, not only the azimuth angle between sender and receiver needs to be computed, but also the elevation angle.

The user may either provide full (three-dimensional) sampling of the antenna to be modeled or samples of the two principal planes only. A method for the interpolation of the whole 3D characteristics based on the principal planes has been proposed in [18] and could be used for the 3D antenna class implementation.

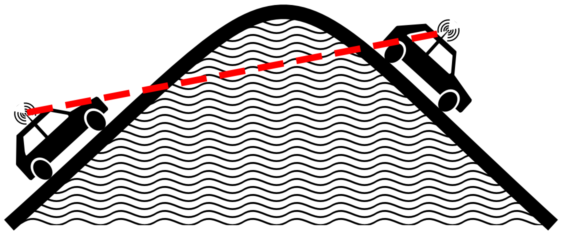


Figure 2. Obstructed communication of two vehicles on a hill.

4) Open Issue: Problems could arise in scenarios featuring hilly terrain. As shown in Figure 2, two cars trying to communicate might be located on opposing sides of a hill.

In a two-dimensional approach, communication might seem easily possible as the absolute distance is smaller than the maximum transmission range and the existence of the hill is not known anyways. A realistic 3D analysis, however, makes it necessary to consider the attenuation caused by the hill itself. Communication might even be completely impossible. Therefore, a solution to properly represent such topographic obstacles is to be developed. Checking the elevation values of a certain number of equidistant points between sender and receiver might be a first approach to tackle this problem.

IV. Three-Dimensional VANET Scenarios

In Section I, we already mentioned an example situation in which a three-dimensional investigation is necessary. In the following, we take a closer look at two other sample scenarios, which would benefit from a holistic three-dimensional simulation.

A. Inter-Vehicle Communication in Parking Garages

A first scenario could deal with the search for a free parking lot in a multi-storey parking garage. Smart parking guidance systems in general are well-known and their realisation based

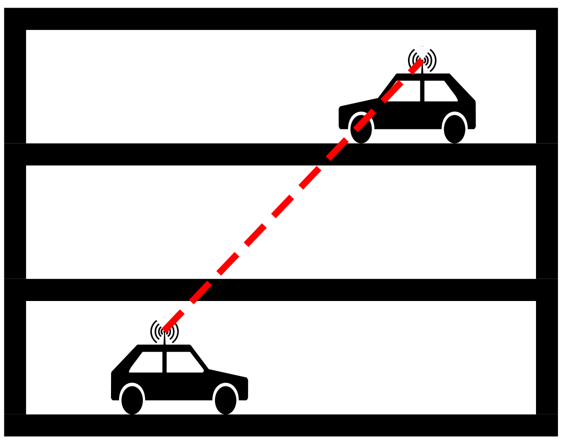


Figure 3. Example scenario of two cars in a parking garage.

on VANETs have already been studied [13], [14]. However, these investigations assume a planar parking area. In the case of a parking garage, one has to deal with a certain number of storeys. Vehicles might try to communicate with other vehicles located on a different level as illustrated in Figure 3.

One can easily guess that such a constellation cannot be simulated in usual two-dimensional simulation environments. Here, the road level or simply the z-coordinate plays a crucial role. Therefore, this scenario needs to be modeled three-dimensionally.

Challenges and open questions are:

- Cars may have to transmit the signal (nearly) straight upwards or downwards. This makes a closer look at antenna patterns very interesting because an elevation angle close to $\pm 90°$ has rarely been investigated in the context of vehicular communications.
- The attenuation induced by the floors of the garage levels must not be neglected. Under certain circumstances it might even be impossible to receive messages from vehicles being located multiple levels above due to the cumulating attenuation of floors in between.
- As an alternative, wire-connected Roadside Units (RSUs) might be installed on each floor. Packet loss and delay can then be compared to the aforementioned variant.

B. 3D Investigation of GLOSA Systems

Another group of interesting scenarios can be characterized by communication with infrastructure, which is usually mounted at higher altitudes. A well-known example are Green Light Optimal Speed Advisory (GLOSA) systems [15]. Here, traffic lights transmit information about the duration of the current phase. Receiving cars approaching the traffic light can then compute the optimum speed in order to prevent unnecessary breaking and accelerating (see Figure 4).

In this scenario, it is important to keep the different heights of transmitter and receiver in mind. The vehicular antenna is mounted at a height of 1.50 m in average, the antenna of the traffic light might be located up to 4.00 m higher. Moreover, an inclined road can further intensify this difference in altitude. The changing relative positions can again have influence on the antenna gains and thus on the maximum transmission range. This in turn can negatively impact the phase prediction. Obviously, this kind of investigation requires a three-dimensional simulation model as the common 2D projection ignores all altitude information.

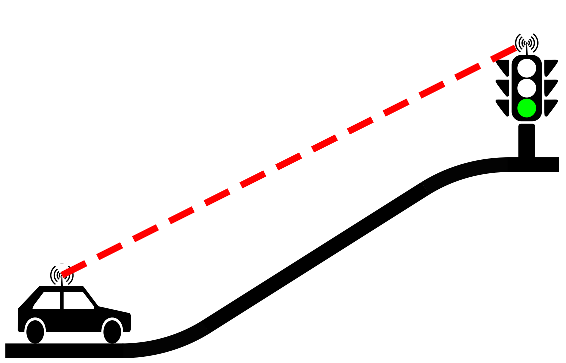


Figure 4. Sample scenario depicting altitude differences in a GLOSA system.

V. Conclusion

In this paper we have pointed out the importance of considering the third dimension in VANET simulation. As opposed to most present simulators, which assume a planar environment, the communicating entities in real-world situations are rarely strictly arranged in a plane. Examples for characteristic scenarios have been presented, summarizing aspects that might lead to significantly differing results.

Thus, several necessary extensions and adjustments towards three-dimensional vehicular network simulation have been outlined. For the traffic simulation, it is important to get detailed altitude data. DEMs can be a possible source for this purpose. In terms of wireless network simulation, this elevation information needs to be utilized accordingly and new obstacle types induced by a 3D consideration have to be dealt with.

Eventually, these improvements shall lead to a more realistic packet-based simulation opportunity to produce more meaningful results for the analysis of VANET protocols and applications.

References

[1] C. Sommer, J. Härri, F. Hrizi, B. Schünemann, and F. Dressler, "Simulation Tools and Techniques for Vehicular Communications and Applications," in *Vehicular ad hoc Networks - Standards, Solutions, and Research*, C. Campolo, A. Molinaro, and R. Scopigno, Eds., Springer, May 2015, pp. 365–392.

[2] C. Sommer, D. Eckhoff, and F. Dressler, "IVC in Cities: Signal Attenuation by Buildings and How Parked Cars Can Improve the Situation," *IEEE Transactions on Mobile Computing*, 13, no., pp. 1733–1745, 2014.

[3] D. Eckhoff, C. Sommer, and F. Dressler, "On the Necessity of Accurate IEEE 802.11p Models for IVC Protocol Simulation," in *75th IEEE Vehicular Technology Conference (VTC2012-Spring)*, Yokohama, Japan: IEEE, May 2012, pp. 1–5.

[4] A. Köpke, M. Swigulski, K. Wessel, D. Willkomm, P. K. Haneveld, T. Parker, O. Visser, H. S. Lichte, and S. Valentin, "Simulating Wireless and Mobile Networks in OMNeT++ – The MiXiM Vision," in *1st ACM/ICST International Workshop on OMNeT++ (OMNeT++ 2008)*, Marseille, France: ACM, Mar. 2008.

[5] C. Sommer, R. German, and F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," *IEEE Transactions on Mobile Computing*, 10, no., pp. 3–15, Jan. 2011.

[6] Q. Lin, C. Li, X. Wang, and L. Zhu, "A Three-Dimensional Scenario Oriented Routing Protocol in Vehicular Ad Hoc Networks," in *2013 IEEE 77th Vehicular Technology Conference (VTC Spring)*, 2013, pp. 1–5.

[7] L. Kristiana, C. Schmitt, and B. Stiller, "Impact of a Three Dimensional Environment to Inter-vehicle Connectivity," in *4th GI/ITG KuVS Fachgespräch Inter-Vehicle Communication*, Berlin, Germany: GI/ITG KuVS, 2016, pp. 1–4. [Online]. Available: http://edoc.hu-berlin.de/conferences/inter-veh-comm-2016.

[8] M. Boban, T. Vinhosa, J. Barros, M. Ferreira, and O. K. Tonguz, "Impact of Vehicles as Obstacles in Vehicular Networks," *IEEE Journal on Selected Areas in Communications*, 29, no., pp. 15–28, Jan. 2011.

[9] M. Boban, R. Meireles, J. Barros, P. Steenkiste, and O. K. Tonguz, "Tall Vehicle Relaying in Vehicular Networks," *IEEE Transactions on Mobile Computing*, 13, no., pp. 1118–1131, 2014.

[10] D. Kornek, M. Schack, E. Slottke, O. Klemp, I. Rolfes, and T. Kurner, "Effects of Antenna Characteristics and Placements on a Vehicle-to-Vehicle Channel Scenario," in *2010 IEEE International Conference on Communications Workshops*, IEEE, Capetown, South Africa, May 2010, pp. 1–5.

[11] L. Reichardt, T. Fügen, and T. Zwick, "Influence of Antennas Placement on Car to car Communications Channel," *3rd Conference on Antennas and Propagation, EuCAP 2009*, no., Mar. 2009.

[12] D. Eckhoff, A. Brummer, and C. Sommer, "On the impact of antenna patterns on VANET simulation," in *2016 IEEE Vehicular Networking Conference (VNC)*, 2016, pp. 1–4.

[13] V. W. s. Tang, Y. Zheng, and J. Cao, "An Intelligent Car Park Management System based on Wireless Sensor Networks," in *2006 First International Symposium on Pervasive Computing and Applications*, 2006, pp. 65–70.

[14] R. Lu, X. Lin, H. Zhu, and X. Shen, "SPARK: A New VANET-Based Smart Parking Scheme for Large Parking Lots," in *IEEE INFOCOM 2009*, 2009, pp. 1413–1421.

[15] R. Stahlmann, M. Möller, A. Brauer, R. German, and D. Eckhoff, "Technical evaluation of GLOSA systems and results from the field," in *2016 IEEE Vehicular Networking Conference (VNC)*, 2016, pp. 1–8.

[16] T. G. Farr, P. A. Rosen, E. Caro, R. Crippen, R. Duren, S. Hensley, M. Kobrick, M. Paller, E. Rodriguez, L. Roth, D. Seal, S. Shaffer, J. Shimada, J. Umland, M. Werner, M. Oskin, D. Burbank, and D. Alsdorf, "The Shuttle Radar Topography Mission," *Rev. Geophys.*, 45, no., 2007.

[17] D. Santos, J. Pinto, R. J. F. Rossetti, and E. Oliveira, "Three dimensional modelling of Porto's network for electric mobility simulation," in *2016 11th Iberian Conference on Information Systems and Technologies (CISTI)*, 2016, pp. 1–6.

[18] N. R. Leonor, R. F. S. Caldeirinha, M. G. Sánchez, and T. R. Fernandes, "A Three-Dimensional Directive Antenna Pattern Interpolation Method," *IEEE Antennas and Wireless Propagation Letters*, 15, no., pp. 881–884, 2016.

Speeding up OMNeT++ Simulations by Parallel Output-Vector Implementations

Stefan Neumeier
Technische Hochschule Ingolstadt
Research Centre
Email: stefan.neumeier@thi.de

Christina Obermaier
Technische Hochschule Ingolstadt
Research Centre
Email: christina.obermaier@thi.de

Christian Facchi
Technische Hochschule Ingolstadt
Research Centre
Email: christian.facchi@thi.de

***Abstract*—Since the complexity of discrete event simulations is growing while the improvements in per CPU performance are at a very low level, there is a need to counteract greatly increasing simulation times. Reducing the execution time can result in simulations that have greater complexity but require only a fraction of time. As simulations write a bunch of data onto the hard disk, improvements in this area may boost up the performance of the simulation. The standard OMNeT++ implementation for writing files onto the hard disk is only single threaded. Hence, adding parallelism to this area may be a step towards the improvement of the runtime of simulations. By applying parallelism to the writing of data, blocking I/O operations can be eliminated as much as possible. This paper presents an approach for writing simulation data asynchronously onto hard disk and demonstrates its capabilities.**

I. Introduction

Nowadays highly accurate simulations are required to provide significant results that are close to reality. The more accurate simulations are, the more input parameters are required. Adding parameters to a discrete event simulation increases its complexity as the additional values usually are used for advanced computations. Hence, the execution time of a simulation is extended. Since the values of these extra parameters are often written onto the hard disk for future analysis, additional time is required not only for computation but also for I/O operations. Usually, simulations can not proceed until the recorded data is entirely written onto the hard disk. This may be a long period of time, depending on the amount of data to be written, the type of the hard drive and its workload. Replacing the deployed hardware with a faster one can lead to reduced execution times. Especially the replacement of a normal hard disk drive (HDD) with a new solid state drive (SSD) boosts the performance of writing data. But if the system is already equipped with a modern SSD as well as other sate-of-the-art hardware and the performance has to be boosted further, replacing the hardware is insufficient. As basic OMNeT++ simulations usually run single-threaded and present CPU manufacturers aim for more cores instead of strongly increasing the single-core performance [7] [11], future hardware improvements will not yield huge speedups for executing simulations. So replacing the CPU may also not help to speed up the execution of simulations.

Thus, boosts in performance will have to be done mainly by changes applied to the software. It has been shown in [6] that minor changes in compiler settings or the use of an alternative malloc/free-implementation can boost up the performance by up to 30 percent. But as already mentioned there, the single core improvements are only a first step and the use of multiple cores has to be applied for further significant improvements. Enabling parallel execution of a simulation is possible as OMNeT++ innately supports this whenever the model meets some criteria like no global variables, static topologies, good partitioned models and a proper lookahead time [13] [9]. As not all models are designed with focus on parallelism or efficient parallel execution by the OMNeT++ standard implementation can not be accomplished, further universal approaches have to be used. One solution to this is the implementation of an efficient and parallel access of the hard disk. The time wasted for waiting on blocking I/O operations can be reduced, which finally leads to a lowered execution time. As usually all scenarios struggle with this problem, the solution is applicable in a wide field.

This paper aims on how to improve the performance of an OMNeT++ based simulation by conducting parallel writing of data onto the hard disk to reduce waiting times on blocking I/O operations. Therefore, Chapter II presents OMNeT++'s approach of writing data onto the hard disk. It is further explained, which kind of data is written during a simulation run and how the writing is usually handled by OMNeT++. It also points out the problems of writing data in parallel. Following, Chapter III shows the OMNeT++ based Veins scenario that was used in this paper. Chapter IV presents a concrete implementation for writing data in parallel. The performance results of the parallel implementation are shown in Chapter V. Finally, section VI sums up the results and gives an outlook on further research topics.

II. Related Background

The OMNeT++ framework provides a built-in way of writing simulation results onto the hard disk. These results are split up into output vectors and output scalars. Output vectors are used to write (changing) values of the simulation over time. Output scalars are used to write summarized results like minimum, maximum or the average of a value of an entire simulation run. Vector data and scalar data is stored in two different files. Due to the often very large vector files, a third file has been introduced in OMNeT++ 4.0. This file is an index file, that enables a faster accessing of the corresponding vector file. Every file usually contains data from one run only. The content is stored in a textual, line-orientated style to allow other third-party tools reading and processing them. [12]

As vector data is written whenever a value changes and scalar data only once per run, vector files are usually bigger than scalar ones. Scalar files have sizes in the range of a few MB, while vector file sizes can grow to several GB and will be written more periodically. Therefore, improvements in the writing of vector files will have a greater impact on the simulation time than improvements in the writing of scalar files.

The default implementation shipped with the OMNeT++ framework provides a sequential and blocking way of writing scalar, vector and corresponding index files. Every time data is written, this default implementation will force the simulation to stop computation and wait until the writing is finished. Long waiting times might be the result, if a lot of data has to be written.

Based on how frequently and how many data is written, there might be huge performance improvements by reducing the waiting times on data to be written. Using a fast SSD might be a suitable solution, but running the Veins beacons scenario on a system that is equipped with a SSD reduced the execution time only about 2 percent compared to the execution on the same system that was equipped with a normal HDD. Thus, another solution for reducing the waiting times on the writing of data has to be found. Asynchronous I/O is part of the modern operating systems and my offers performance boosts when a lot of data has to be written [5] [1]. Hence, it is investigated if applying asynchronous writing of data to OMNeT++ also helps to improve the performance.

The test system used in this paper consists of an Intel(R) Xeon(R) E5-2620 v2 (2.10 GHz) CPU. The system has 8 Gb of RAM built in and is equipped either with a HDD or a SSD. The version of OMNeT++ is 4.6 and the version of Veins is 4a1 with the INET MAC layer. Sumo is in version 0.21.

Disabling the writing of vector data considerably reduces the execution time of a simulation. The execution time of the OMNeT++ based Veins scenario in the beacons configuration is at about 210 seconds when simulating 1044 seconds. By disabling the writing of vector data in this scenario, the execution time is reduced by about 22 percent down to 164 seconds. Even if the system is equipped with a SSD, the execution time can be reduced by about 20 percent when no vector data is written. These values can be seen as the theoretical maximum that might be reached by parallel writing implementations in this scenario.

Improving the performance of the execution can be done by adding a new thread, which writes data in parallel to the computation of the simulation. Thus, the computation of the simulation can proceed without waiting on the hard drive. Figure 1 shows the execution of a simulation with two threads in parallel. In comparison to the execution with one thread there is no time lost for waiting on the hard drive. The red line indicates spots where the simulation can proceed even if data is written onto the hard disk and hence represents the time saved by not waiting on I/O tasks.

Unfortunately, the normal execution is usually not finished if all computation is done. It continues running until all remaining data is stored on the hard drive. Figure 2 represents the usual behavior of data during the execution of the simulation. The initializing and end of the simulation are symbolized through the vertical lines. The arrow symbolizes the creation

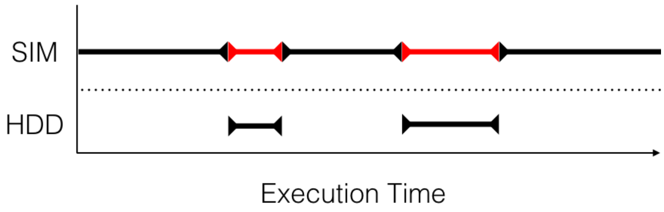


Fig. 1. Illustration of the computation of the simulation and the writing of data in two different threads.

of a node, while the star symbolizes its destruction. The line between the arrow and the star, that looks like saw teeth, represents the amount of data which is remaining to be written per node. During the simulation, the amount of data remaining to be written increases. Every time the writing is triggered, all data of the specific node is written. Now, problematic data is data, that is generated but will not be written in parallel to the computation. *Node n* is destroyed before the whole simulation ends. Symbolized by the read red rectangle, its remaining data can be written in parallel to the computation as the simulation is still running. The situation in *Node 0* is different. This node is destroyed at the end of the simulation. Thus, its remaining data can not be written in parallel, as the computation has already finished. Marked by the green rectangle, the writing is sequential again and will not lead to a reduction of the execution time.

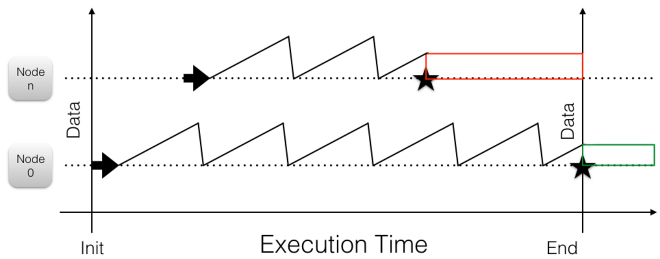


Fig. 2. Sequence of a simulation. Nodes are created (arrow) and destructed (star) at various times of the simulation. Data is generated and written during the execution (red) of the simulation and afterwards (green).

III. Veins-Based Beacons Scenario

In general, nearly all OMNeT++ scenarios may benefit of writing data in parallel. Depending on the CPU usage of the scenario and the amount of data to be written, the level of improvement differs. In this paper, the Veins-based beacons scenario is investigated. This scenario consists of cars that are driving on a map of the German city Erlangen. They are frequently sending messages, as usual in Car2X scenarios [8] [10]. Due to the dynamic character of the scenario, nodes are created and destructed dynamically.

The only simulation parameter of the Veins scenario that changes in this paper is the simulation time, which is set to 600, 900 or 1044 seconds. Table I shows the different vector file sizes based on these configurations. The file size of the Veins beacons scenario ranges from about 915 MiB in a test interval of 600 simulated seconds up to about 1416 MiB for 1044 simulated seconds.

TABLE I. DIFFERENT VECTOR FILE SIZES BASED ON DIFFERENT SIMULATION TIMES OF THE VEINS BEACONS SCENARIO.

Simulation time (s)	File Size (MiB)
600	914.86
900	1381.19
1044	1415.55

IV. IMPLEMENTATION

The implementation of the parallel writing of data is done as an extension to the OMNeT++ framework. Due to its extensibility, the core of OMNeT++ will not have to be changed. The solution shown in this paper does not use the asynchronous I/O APIs offered by the operating systems. These APIs are different on the various systems. Using them would lead to unwanted dependencies.

For obtaining the file format, the index file based implementation provided by OMNeT++ is used as a basis. It is extended by applying it to a new thread which writes the vector data in parallel to the computation of the simulation. All vector data is written in parallel, except the basic data at node creation. This data is written by the main thread, which also executes the simulation. This solution has been chosen due to the good ratio between implementation effort and performance outcome.

The implementation introduced in this paper has a global buffer for exchanging data between the data generating nodes and the writing thread. This global buffer is a First In First Out (FIFO) queue to ensure the correct order of written data. Every single nodes has its own local buffer. The local buffer contains the simulation data, which is generated per node. If the local buffer is full, a new one is created and the old one is passed to the global buffer. This is done by passing the address of the corresponding memory to the global queue. Copying of memory is avoided for achieving the maximum performance. The local buffer has an option for configuring the maximum buffer size. This feature enables the control of memory consumption. It also helps to increase the performance of finally writing data onto the hard disk by determining the size of blocks being written in a sequence.

The parallel writing thread takes the oldest entry of the global queue. It removes the entry, determines the contents of the corresponding buffer, writes its contents onto the hard disk and frees the memory. All these tasks can be done independent of the proceeding of the simulation. Only the removing of an entry requires secure access to the queue, which may block the queue for the simulation and thus reduce the performance. The relation between global buffer, local buffer and the writing thread is shown in Figure 3. This setup helps to speed up the execution as the new local buffer can be filled in parallel to the old one being written by avoiding huge copy operations of data.

The buffers are implemented as a queue to provide the required FIFO functionality to keep the data, which has to be written, in correct order [4]. A benefit of using a queue is the constant time consumption for insertion and deletion of elements, as elements at the front and end can be accessed directly [2].

The implementation has a writer thread, that is usually in sleep mode. It is woken up using condition variables whenever

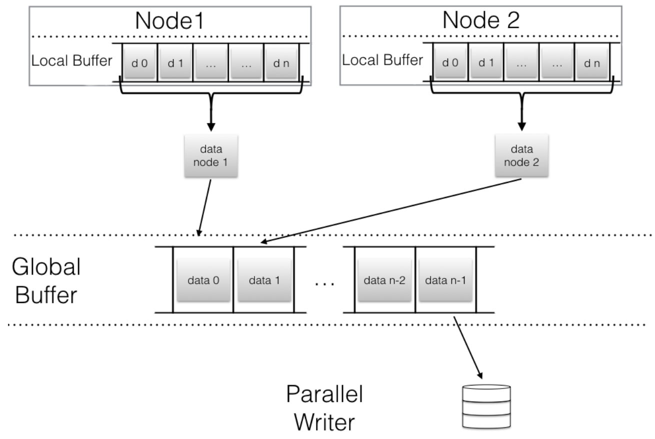


Fig. 3. Relation between the local and global buffer. Data is first saved in the local buffer of each node. If the local buffer is full, its pointer is transferred to the global buffer and written in parallel by the writing thread.

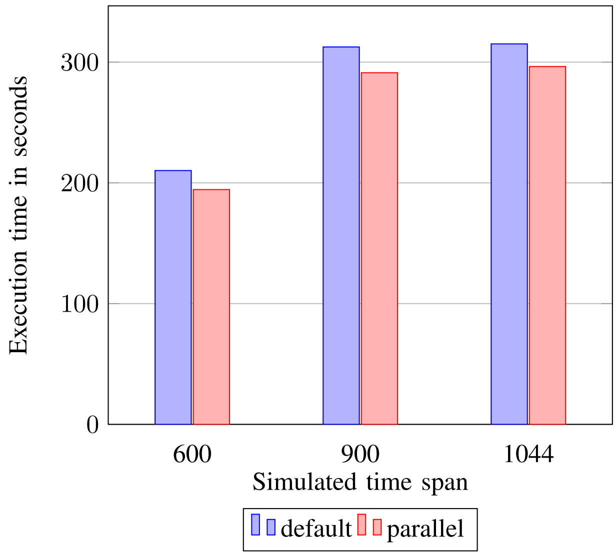
data is available in the global buffer. If all data of the global buffer is written, the thread is forced to sleep again. In general, more solutions have been tested. A lock-free-queue and a constant pulling instead of condition variables also have been investigated. Showing all thes combinations would reduce the legibility of this paper drastically. Thus, only the best performing solution is presented in this paper.

V. MEASUREMENTS

For being able to compare the basic OMNeT++ implementation with the parallel one, both solutions were compared directly. The buffer size for the default OMNeT++ vector writing implementation has been set to default buffer. Thus, data is only written whenever the simulation ends or the limit of 16 Mb of memory consumption is reached.

For having an equal comparison, the parallel implementation is represented by the mean of its execution time with local buffer size is set to 500 and 1000, having not too many single writing events. Simulation runs with smaller buffer sizes, e.g. 50, were tested, but not used as they have provided a very poor performance. In case of a buffer size of 50 the implementation was up to two times slower than the mean of 500 and 1000 combined. The smaller the buffer size is the more frequently the writing is triggered. Every writing process has to call system events, which leads to additional waiting times that lead to a higher total duration of the writing process. Consequential, the writing can not finish before the simulation ends and thus the system falls back into sequential mode with no performance benefit. Storage mediums usually perform faster if they are able to write a bunch of data in a sequential way [3] and not small pieces of data in high frequency. Additionally, every triggered writing to the vector file leads to a new entry in the corresponding index file. By using the scenario described above and setting a buffer size of 50, the index file grows up to about 386.5 MiB. This is nearly ten times more than compared to a buffer size of 500, where it uses about 39.6 MiB. Thus, better performing behavior can be enforced by setting a bigger buffer size. But if the buffer size is selected too huge, too many data remains to be written at the end of the simulation. This again leads to a reduced performance.

Fig. 4. Comparison of the default and parallel implementation of the Veins scenario on the test system with a HDD.



For performance measurements, the beacons configuration of the Veins scenario was used. This scenario was executed on the test system with a normal HDD and a SSD. Figure 4 shows the performance of the default implementation compared with the parallel one on a system with a normal HDD. The parallel implementation is about 7 percent faster than the normal implementation.

On a system equipped with a SSD the parallel implementation is also 5 to 7 percent faster compared to the default OMNeT++ implementation. The CPU utilization was increased by about 17 percent to 109.

VI. Conclusion

The results shown in this paper are a satisfying approach for improving the execution time of an OMNeT++ based simulation by applying a thread to write data in parallel onto the hard disk.

The improvements shown in this paper range from about 5 to 7 percent depending whether a SSD or a HDD is installed. A synthetic benchmark using a modified version of the basic OMNeT++ scenario ”TicToc” has also been conducted. This scenario was modified to constantly write a huge amount of data and to conduct computations (multiple random number generations). Different configurations with a different number of nodes, values to be written per node and simulation times showed an improvement of up to 25 percent when writing data in parallel.

Parallel writing of data will help to improve the performance of a simulation, but the improvements highly depend on the scenario. A CPU bound scenario with a plenty of data to be written will benefit more from writing data in parallel than a CPU bound simulation with less data to be written. A mainly I/O bound scenario will benefit less, if the CPU is not really utilized and the computation ends quickly. Parallel writing can improve the execution time only if the simulation has not finished its computation as long as data still has to be written. If the simulation is done with its computation and there is data left to be written, this is done in a sequential way. The consequently means that there are no improvements at this time.

Future work will provide a more powerful and efficient implementation to further reduce the execution time. Steps to provide this consist of a solution, where only the writing thread accesses the vector file. The implementations presented in this paper allow access to the vector file by the writing thread and the main thread, which leads to preventable file-locks and resulting waiting times.

Finally, one should consider, that even if parallel writing of data onto the hard disk improves the performance, further steps on the individual scenario are required to achieve a higher overall performance. These steps mainly consist of applying parallelism to the simulated scenario itself to enable a parallel computation. Thus, writing data in parallel may only be an additional step.

References

[1] Suparna Bhattacharya, Steven Pratt, Badari Pulavarty, and Janet Morgan. Asynchronous I/O support in Linux 2.5. In *Proceedings of the Linux Symposium*, pages 371–386, 2003.

[2] Nell B Dale and Susan C Lilly. *Pascal plus data structures, algorithms, and advanced programming*. Jones & Bartlett Learning, 1995.

[3] Intel. Intel Solid-State Drive 520 Series - Product Specification, February 2012.

[4] DS Malik. *data structures using C++*. Cengage Learning, 2009.

[5] Microsoft. *Synchronous and Asynchronous I/O*. Microsoft.

[6] Raphael Riebl, Stefan Neumeier, and Christian Facchi. Inter-Vehicle Communication on the Run - Experiences From Tweaking Veins Runtime Performance. In *Proceeding of the 3rd GI/ITG KuVS Fachgesprach Inter-Vehicle Communication (FG-IVC2015)*, 2015.

[7] Phil Rogers. The Programmers Guide to the APU Galaxy. AMD Fusion Developer Summit, Jun 2011.

[8] J. Santa, F. Pereniguez, A. Moragon, and A.F. Skarmeta. Vehicle-to-infrastructure messaging proposal based on CAM/DENM specifications. In *Wireless Days (WD), 2013 IFIP*, pages 1–7, Nov 2013.

[9] Y Sekercioglu, András Varga, and Gregory Egan. Parallel simulation made easy with omnet++. In *The European Simulation Symposium (ESS 2003)(Alexander Verbraeck 26 October 2003 to 29 October 2003)*, pages 493–499. SCS European Publishing House, 2003.

[10] C. Sommer, R. German, and F. Dressler. Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis. *Mobile Computing, IEEE Transactions on*, 10(1):3–15, Jan 2011.

[11] Herb Sutter. The free lunch is over: A fundamental turn toward concurrency in software. *Dr. Dobb's journal*, 30(3):202–210, 2005.

[12] András Varga. *OMNeT++ User Manual*. OpenSim Ltd., 4.6 edition, 2014.

[13] András Varga and Rudolf Hornig. An overview of the OMNeT++ simulation environment. In *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, page 60. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008.

Towards CityMoS: A Coupled City-Scale Mobility Simulation Framework

Daniel Zehe*, Suraj Nair*, Alois Knoll*† and David Eckhoff*
* TUMCREATE, 1 CREATE Way, #10-02 CREATE Tower, Singapore 138602
† Technische Universität München
{firstname.lastname}@tum-create.edu.sg, knoll@in.tum.de

***Abstract*—Simulation has become one of the primary tools for the evaluation of transportation systems. To investigate novel concepts such as autonomous vehicles or car-2-car communication, new simulation and modelling techniques are required to increase the meaningfulness and impact of simulation studies. To this end, we develop CityMoS, an easy-to-use, city-scale, microscopic, agent-based mobility simulation framework. The main intended use of this framework is the assessment of autonomous vehicles (AV) with a particular focus on the interaction with conventional vehicles (CV). In order to achieve this, we aim at revising conventional car-following models as well as introducing novel hierarchical models for AVs. A focus will lie on providing a validated city-scale scenario that researchers can parametrise and use to evaluate novel applications. Lastly, we put an emphasis on making our simulation environment interoperable with existing tools either by coupling them or by providing open interfaces to interact with CityMoS.**

I. Introduction

The future of mobility will most likely be autonomous. This does not only add challenges to private road traffic, but also to public transportation on and off the road. In order to evaluate the impact of such fundamental changes on current and future infrastructure, simulation tools are required that allow researchers and policy makers to evaluate possible scenarios to support long-term decision making. Ideally, these tools are able to simulate entire cities or regions with a large number of agents (>100k) while at the same time maintaining a manageable level of complexity.

The number of agents is by far not the only requirement: The performance of the simulation environments has to be magnitudes better than wall-clock time in order to explore the large parameter spaces associated with autonomous vehicles. Also, most existing microscopic mobility models are not sufficient to simulate autonomous vehicles (AV) as they only focus on the low-level control of the vehicle without considering trajectory planning or higher level decision making. Lastly, in the context of intelligent transportation systems and smart cities, the high level of connectivity will have a significant impact on the traffic. This includes other participants, e.g., pedestrians, as well as infrastructure such as car parks, signage or traffic lights. Disregarding this aspect will therefore not allow researchers to investigate the full scope of future traffic.

A new approach towards city-scale mobility simulation is required, where many different concepts of models have to be integrated. The simulation has to provide the possibility to explore large city areas, while at the same time offering a high level of detail. In this paper, we are presenting the current plans for city-scale mobility simulation platform *CityMoS*, which aims at solving the aforementioned challenges. An integral part of this simulation platform is the support for high performance computing environments and cloud-based solutions. This is necessary due to the computationally intensive nature of high-detail city-scale simulations that might be infeasible to run on a workstation computer.

To achieve this high level of detail and to arrive at meaningful results, a focus of CityMoS will be the modelling of conventional vehicles and autonomous vehicles. Microscopic mobility models for human drivers are optimistic in the way that they do not allow for human errors or unexpected behaviour. This is, however, necessary to learn how, in a mixed traffic scenario, autonomous vehicles will react to such behaviour. Secondly, limiting autonomous mobility to only car-following and lane-change models is insufficient as it disregards higher level control of the vehicle that will have a decisive impact on the decisions and routing of the vehicle. Lastly, the interconnectivity of intelligent transportation systems has to be taken into account. To this end, we implemented the TraCI protocol into CityMoS to allow coupling with network simulations such as ns-3 [1] and OMNeT++ [2].

II. State of the Art

There exists quite a number of traffic simulation platforms.

Among the most widely used commercial traffic simulators, VISUM and VISSIM [3] play an important role. VISUM is a macroscopic traffic simulator that focuses on the modelling of traffic demand and traffic flows. VISSIM on the other hand simulates on a microscopic level, that is, each vehicle is an agent with specific behavioural models such as car-following and lane-change models. Due to the closed-source nature of VISSIM, it is not possible to extend it in a way so that studies of mixed CV and AV traffic can be conducted. VISSIM is therefore more suitable for traffic planning in general than for the development of models to study of novel applications.

The *Multi-Agent Transport Simulation* (MATSIM) [4] offers a microscopic traffic simulation platform for the simulation of large scenarios. It works on activities that are fixed at the beginning of the simulation, which makes it impossible to change routes or behaviour of vehicles during run-time. This is a major drawback when it comes to investigation of autonomous mobility in an interconnected intelligent transportation system.

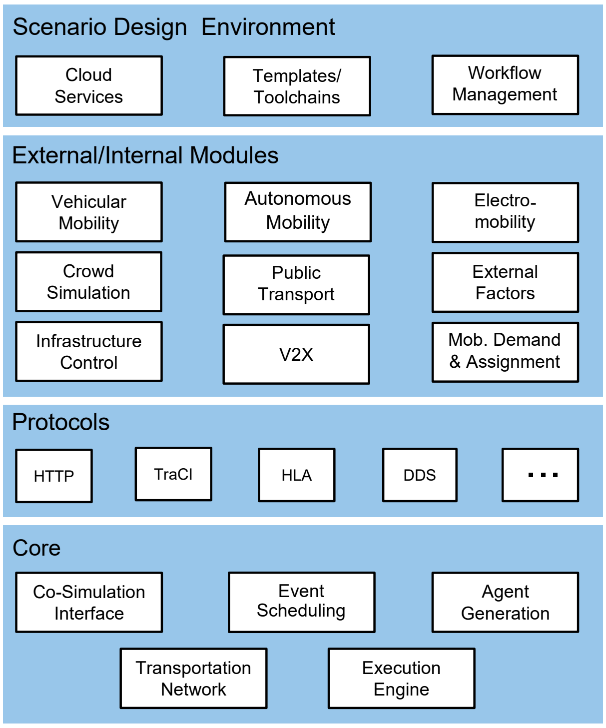


Fig. 1. Components of CityMoS.

Furthermore, creating scenarios in MATSIM is a complex task due to the lack of a general user interface.

SUMO, short for *Simulation of Urban Mobility* [5], is a microscopic agent-based traffic simulator that allows to simulate a large number of vehicles in a predefined network. It features well-known car-following models such as the intelligent driver model (IDM) [6], but lacks the possibility to easily extend or introduce new models, which is something we believe is necessary to study autonomous vehicles. The road network can be generated from open street map data or it can be synthetically generated. SUMO comes with a graphical user-interface but only for controlling the simulation and inspecting the current simulation progress, the configuration has to be created externally and loaded into the GUI at start. SUMO comes with TraCI, an interface to retrieve information about the simulation and also to control it during simulation time. This interface is based on TCP socket and a binary protocol. It has been used to bidirectionally couple network simulators such as ns-2 (Trans [7]), ns-3 (iTetris [8]) or OMNeT++ (veins [9]) with SUMO. This is important to learn how network communication can influence traffic and also how the network topology affects communication metrics.

A simulation environment that combines various simulations from different domains is VSimRTI [10]. It couples various simulators such as OMNeT++, VISSIM, or SUMO following concepts similar to IEEE HLA [11]. The simulator itself takes the role of the controller, dealing with synchronization, interaction and all other management tasks. Other simulators can be coupled using generic VSimRTI interfaces. The concepts and methods of VSimRTI are interesting and will serve as valuable input for the development of CityMoS.

In earlier work, we presented SEMSim (*Scalable Electromobility Simulation* traffic [12]), a (sub-)microscopic traffic simulation platform initially developed to study the effect of electric vehicles on the transport system. In this context, submicroscopic means that, in addition to lane-change and car-following models, agents also incorporate different vehicle components such as the battery. SEMSim is designed to be a city-scale simulation platform that can be used in a single or multi-node execution environment. It is the basis for the development of CityMoS.

III. CityMoS

The *City Mobility Simulator* (CityMoS) aims at being an extendible simulation platform that allows the evaluation of various mobility aspects holistically on a city scale. An overview of all planned and already functional components is given in Figure 1. The core is a discrete event simulation (DES) platform that deals with the dispatching and scheduling of events as well as the necessary steps to realize parallel execution. A particular focus will be set on the development of new modules to incorporate more modes of transport as well as the support for various co-simulation protocols to couple other simulation environments with CityMoS. This includes the TraCI protocol which is widely used in the context of vehicular network simulation [9]. Furthermore, we aim at supporting HLA as well as being Functional Mockup Interface (FMI) [13] conformable.

Microscopic mobility models: Apart from simulating private vehicle traffic, we aim to also integrate public transport (including non-road-based transport), as well as pedestrian movement. The main advantage of using CityMoS over other solutions is the integration of autonomous models for personal vehicles and also for road-based public transport vehicles. To this end, we develop a hierarchical model that is capable of incorporating various types of incoming information, planning routes and trajectories, instead of solely controlling low-level actions of the vehicle. In order to study the effects autonomous vehicles have on conventional human-driven vehicles, we further need to revise existing car-following and lane-changing models as the ones that are currently used in simulation tools have several drawbacks: They are deterministic and collision-free, which means that unpredictable or even irrational human behaviour will not be represented in the simulation. It is however exactly this kind of behaviour which may be of interest when studying mixed traffic of autonomous and conventional vehicles. The introduction of accidents occurrence model capability will further allow to simulate traffic flow control strategies.

Scenario modelling: We aim at extending the functionality of CityMoS by including multi-mode traffic assignment and demand models. This is required since SEMSim worked solely on tempo-spatial origin-destination pairs for private vehicular traffic. CityMoS, however, with the inclusion of public transport, pedestrian and autonomous mobility, needs demand modelling on a different level. Since this data is not trivial to obtain for all scenarios, work in CityMoS will also go into building tools and guidelines to create such information. Additionally, the goal is to provide realistic large-scale multi-modal demand input data for an easy start into using CityMoS for all kind of simulation scenarios. Ideally, CityMoS will come with calibrated and validated scenarios consisting of road networks and travel demand. This enables researchers to quickly set up their simulation without spending time going through the complex process of road network conversion, demand modelling and general data-preprocessing steps.

Bidirectionally coupling: Making CityMoS capable of coupling with existing simulation environments and allowing for the easy integration of other models is one of the key objectives. We are in the process of implementing the TraCI protocol [14] into CityMoS to allow interaction with OMNeT++ and other TraCI-compatible simulators. This opens the door for the evaluation of communication networks, or in a broader scope, smart city applications. CityMoS will then serve as the provider of mobility information, offering information about the states of the simulated agents. At the same time, this interface can be used to control agent behaviour in the simulation. Coupling always comes with overhead as information between the simulations has to be exchanged and synchronization has to be taken care of. A clear software architecture with pre-defined interfaces will help users to implement their own models into CityMoS to avoid this overhead.

User interface: Another important aspect to increase user acceptance is to provide a reliable and well-structured user interface. The visualisation of the simulation for demonstration purposes needs to be appealing and meaningful, as showing how a certain application affects the simulation is often a decisive factor for research. User interfaces go beyond only visualising the current state of the simulation and includes pre-simulation data-processing tools, configuration tools as well as model and parametrization repositories, and a post-simulation analysis. We want to make the set-up process for simulations as self-explanatory and guided as possible while still providing full control over all simulation parameters.

Performance: We are aiming to provide a cloud-based solution for large-scale experiments as well as a local execution environment for smaller studies. Apart from modelling components and entities in the simulation environment, the execution of the simulation studies conducted with CityMoS should be faster than wall-clock time for a mega-city application. This means we are working on the execution environment to be scalable from a workstation computer for small-scale experiments up to high performance distributed computing infrastructures. To further speed up the simulation, we are looking into possibilities to facilitate specialized co-processors like FPGAS or GPUs.

IV. Conclusion

In this paper, we presented the current state and the road map for the development of CityMoS. It aims to be an easy-to-use city-scale mobility simulator for both decision makers and experts from academia and industry to allow high resolution modeling of various participants in an intelligent transportation system. It allows parallel, high performance simulation of time horizons ranging from minutes up to multiple days and combines various aspects of transport systems to allow users to draw holistic conclusions. We put a strong focus on the interoperability with other simulation environments, e.g., to allow studies regarding inter-vehicle communication and autonomous mobility.

Acknowledgements

This work was financially supported by the Singapore National Research Foundation under its Campus for Research Excellence And Technological Enterprise (CREATE) programme.

References

[1] G. F. Riley and T. R. Henderson, *The ns-3 Network Simulator*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 15–34.

[2] A. Varga, "The OMNeT++ Discrete Event Simulation System," in *European Simulation Multiconference (ESM 2001)*, Prague, Czech Republic, June 2001.

[3] M. Fellendorf and P. Vortisch, "Microscopic Traffic Flow Simulator VISSIM," in *Fundamentals of Traffic Simulation*, ser. International Series in Operations Research & Management Science, J. Barceló, Ed. Springer, 2010, vol. 145, pp. 63–93.

[4] *The Multi-Agent Transport Simulation MATSim*. London: Ubiquity Press, 2016.

[5] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO - Simulation of Urban MObility," *International Journal On Advances in Systems and Measurements*, vol. 5, no. 3&4, pp. 128–138, December 2012.

[6] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical Review E*, vol. 62, pp. 1805–1824, Aug. 2000.

[7] M. Piorkowski, M. Raya, A. L. 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, pp. 31–33, 2008.

[8] R. Bauza, J. Hernandez, J. Gozalvez, S. Vaz, D. Valiente, I. Aguado, and A. Gonzalez, "iTETRIS Heterogeneous Wireless Communication Platform for the Large-Scale Evaluation of Cooperative Road Traffic Management Policies," in *9th International Conference on ITS Telecommunications (ITST 2009)*, Lille, France, October 2009.

[9] 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, January 2011.

[10] B. Schünemann, "V2X Simulation Runtime Infrastructure VSimRTI: An Assessment Tool to Design Smart Traffic Management Systems," *Elsevier Computer Networks*, vol. 55, no. 14, pp. 3189–3198, October 2011.

[11] J. S. Dahmann and K. L. Morse, "High level architecture for simulation: An update," in *Proceedings of the Second International Workshop on Distributed Interactive Simulation and Real-Time Applications*, ser. DIS-RT '98. Washington, DC, USA: IEEE Computer Society, 1998, pp. 32–. [Online]. Available: http://dl.acm.org/citation.cfm?id=789095.790716

[12] D. Zehe, A. Knoll, W. Cai, and H. Aydt, "{SEMSim} cloud service: Large-scale urban systems simulation in the cloud," *Simulation Modelling Practice and Theory*, vol. 58, Part 2, pp. 157 – 171, 2015, special issue on Cloud Simulation.

[13] M. Otter, T. Blockwitz, E. Hilding, A. Junghanns, J. Mauss, and H. Olsson, "Functional mockup interface overview," Jan 2010.

[14] 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 11th communications and networking simulation symposium*. ACM, 2008, pp. 155–163.

Proceedings of the 5th GI/ITG KuVS Fachgespräch Inter-Vehicle Communication (FG-IVC 2017)
Anatoli Djanatliev, Kai-Steffen Hielscher, Christoph Sommer, David Eckhoff, and Reinhard German (Eds.)

On the Feasibility of Multi-Hop Communication in a Realistic City Scenario

Karsten Roscher
Fraunhofer Institute for Embedded Systems and Communication Technologies ESK, Munich, Germany
Email: karsten.roscher@esk.fraunhofer.de

***Abstract*—Vehicular ad hoc networks (VANETs) are about to enter the market in the upcoming years. While direct message exchange is limited to a few hundred meters and mostly line-of-sight, multi-hop forwarding can significantly extend the communication range. In this paper, the influence of network density and vehicle distribution on multi-hop communication in VANETs is investigated using a model of the city of Luxembourg. Introducing an idealistic reference routing we identify the need for high penetration rates to achieve acceptable reliability independent of the selected routing protocol. A comparison with Greedy forwarding shows its weaknesses to select proper next hop candidates when the number of equipped vehicles increases.**

I. Introduction

Wireless information exchange between vehicles is envisioned to improve traffic safety and efficiency with an extended perception range beyond on-board sensors. Furthermore, cooperation and strategy alignment will play a crucial role for highly automated driving. However, the communication range of vehicular ad hoc network (VANET) technologies like 802.11p [1] and ETSI ITS-G5 [2] is limited to a few hundred meters. Large obstacles like buildings or vehicles reduce it even further.

Multi-hop communication can increase the available range by utilizing intermediate nodes to relay the packet on behalf of the source. Yet, selection of relay nodes and thus forming a forwarding path remains a challenging tasks due to constantly changing network topologies. Geographic routing protocols appear to be a promising candidate for multi-hop communication in VANETs [3] and are also part of the ETSI ITS framework [4].

In geographic routing, the location of communicating nodes is used for addressing and forwarding decisions. A *next hop* is usually selected on demand based on information collected via periodic status messages, e.g. network beacons or Cooperative Awareness Messages (CAMs) [5]. Convergence of the process is ensured by only selecting nodes geographically closer to the destination than the current position. A variety of algorithms was suggested in recent years, a comprehensive overview can be found in [6]. *Greedy Perimeter Stateless Routing* (GPSR) [7] is one of the few examples providing a fall-back strategy to route around *dead ends* by temporarily considering nodes that are not closer to the destination. However, this so-called *Perimeter Mode* was not included in the ETSI GeoNetworking standard [8].

Reliable multi-hop communication highly depends on sufficient network density: If there are no suitable candidates for relaying, forwarding is impossible. To avoid the negative impact of insufficient network connectivity, protocols in VANETs are often evaluated with well populated scenarios - likely a straight highway or a Manhattan Grid. However, the minimum requirements for multi-hop communication in realistic deployment scenarios remain unknown.

In this paper, a model of the city of Luxembourg [9] is used to investigate the feasibility of multi-hop communication for varying traffic densities and technology penetration rates. An idealistic reference routing protocol is introduced to investigate forwarding limits posed by connectivity of the network and the distance convergence requirement of most geographic protocols. The optimal protocol is compared to the *Greedy* algorithm standardized in [8].

The remainder of this work is outlined as follows: The reference routing algorithm is introduced in Section II. Section III describes the investigated scenario in more detail. Evaluation results are provided in Section IV, while Section V concludes the paper with an outlook on topics for future work.

II. Reference Routing with a Deterministic Communication Model

The performance of a specific routing algorithm depends on the proper selection of forwarding candidates as well as the network topology, i.e distribution and density of direct neighbors. To investigate these issues separately we introduce a reference routing algorithm that finds optimal paths under the constraints of a simplified deterministic communication model. It leverages perfect information about all nodes in the simulation environment and is therefore only intended for comparison with existing protocols and analysis of scenarios. In contrast to the work provided in [10] we focus on the achievable performance using a global view on the network not limiting the selection to a local strategy. Since all possible paths are investigated, we will also find those that may require suboptimal choices at intermediate hops.

A. Communication Model

Models for wireless communication links in network research usually include stochastic components to model fading characteristics and other random properties of the channel. However, in order to do an exhaustive search for shortest paths a deterministic model is required to describe under which conditions two vehicles can communicate using a direct link. Since we intend to compare the reference routing with results obtained from real protocol implementations, we directly link our model to the propagation characteristics of simulated channel. However, only the deterministic part is used neglecting signal variations introduced by fast fading etc.

The evaluation in Section IV is based on the channel model derived from measurements by Cheng et al. [11] extended with support for buildings as obstacles [12]. According to the model, two nodes are directly connected, if the received power is above a configurable threshold p_{rx_min} which depends on the data rate and specific receiver characteristics. In this paper, we assume $p_{rx\_min} = -91\ dBm$ based on the suggestions in [13].

B. Shortest Path Optimal Routing

Assuming global knowledge within the simulation environment, a shortest path with respect to the number of hops can be found with the following algorithm:

Require: source s, destination d, set of all nodes N,
Initialize connectivity tree T with root s
$C = N \setminus \{s\}$, $L_{in} = \{s\}$, $L_{out} = \emptyset$
while $C \neq \emptyset$ **and** $L_{in} \neq \emptyset$ **do**
  for all $c \in C$ **do**
    for all $l \in L_{in}$ **do**
      if $connected(c, l)$ **then**
        $C = C \setminus \{c\}$
        $L_{out} = L_{out} \cup \{c\}$
        Add c as child of l in T
        if $c = d$ **then**
          return Path from s to c in T
        end if
      end if
    end for
  end for
  $L_{in} = L_{out}$, $L_{out} = \emptyset$
end while
return No path

The outlined algorithm considers all remaining nodes in each step independent of their location with respect to the destination. This strategy is called *select all*. To model the distance convergence of most geographic routing protocols, we define a second strategy *select closer* where children have to be closer to the destination than their parents in the tree. It limits the paths found by the reference algorithm to those discoverable by protocols requiring constant progress towards the destination. The difference between both strategies shows the potential packet loss introduced by local routing optima.

III. Scenario

A. A Realistic City Scenario

The *Luxembourg SUMO Traffic (LuST) Scenario* [9] provides 24 hours of mobility simulation for the city of Luxembourg using the traffic simulator SUMO [14]. It also includes the outlines of buildings used by the propagation model. Fig. 1 shows the number of active vehicles over an entire day. We selected three different times of the day to cover varying traffic densities as summarized in Table I.

TABLE I
SIMULATION SCENARIOS

Scenario	Time	Vehicles
low	11:05am	1700
medium	1:20pm	3200
high	8:15am	5000

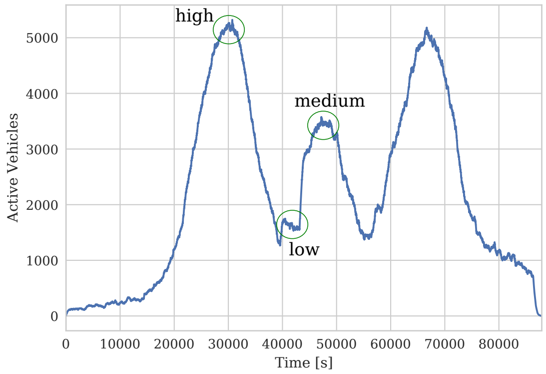


Fig. 1. Active vehicles in the LuST scenario over 24h and the selected periods.

B. Communication Patterns

Communication patterns characterize the data flow within the network: which nodes communicate and how often do they exchange packets. The patterns directly depend on the specific applications. However, with a lot of research focusing on broadcast dissemination of information in VANETs, real applications using unicast communication between vehicles are still missing. We therefore model the data traffic according to the following principles:

1) Communication is limited to a local context with information exchange in close proximity happening more frequent.
2) Communication is sporadic with only a few exchanged messages per *session* in contrast to stream-like data flows in consumer networks dominated by multimedia traffic.

In our evaluation, each *session* consists of a single packet. First, a source vehicle s is selected randomly among all active vehicles v_i in the simulation. Then, a random target distance d_t is drawn from an exponential distribution with a mean of 500 m and a maximum bound at 2000 m. Finally, the selection of the destination is based on minimizing the difference between d_t and $dist(s, v_i)$.

IV. Evaluation

A. Simulation Environment and Parameters

The simulation environment consists of the network simulator ns-3 [15], the traffic simulator SUMO [14] and the ezCar2X framework [16], [17] implementing the ETSI ITS protocol stack. Multiple simulations of the same parameter set with different random seeds were run using [18]. The main

TABLE II
SIMULATION PARAMETERS

Parameter	Value
Standard	802.11p
Tx Power, Tx Rate	23 dBm, 6 Mbps
Propagation Model	Cheng et al. [11] with [12]
Packet Size, Packet Interval	300 Bytes, 0.1 s
Beaconing Scheme	Adaptive CAM generation [5]
Neighbor Timeout	5 s
Simulation Runs	32
Duration per Run	1000 s

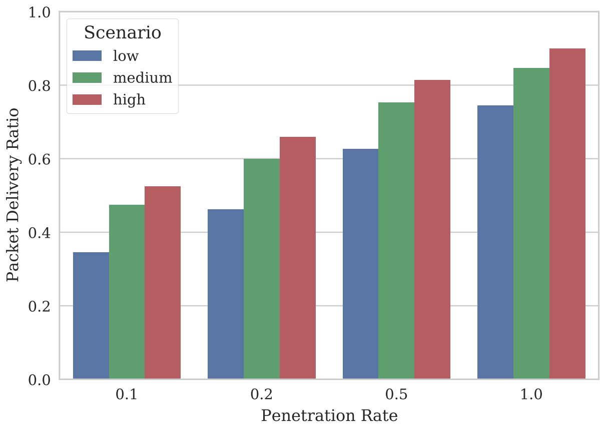


Fig. 2. Packet delivery ratio with the *select closer* strategy.

simulation models and parameters are summarized in Table II; beaconing scheme and neighbor timeout only apply to Greedy Forwarding.

B. Reference Routing

The packet delivery ratios achieved by applying the reference routing algorithm with the *select closer* strategy are summarized in Figure 2. Each scenario was simulated with 4 different penetration rates: 10%, 20%, 50% and 100%. As expected, the delivery ratio increases if more communicating vehicles are available. This is consistent across scenarios as well as penetration rates.

With only 10% equipped vehicles the expected delivery ratio exceeds 50% only during morning rush hour. Of the successfully delivered packets, roughly 75% (i.e. 37.5% of all transmitted packets) can be attributed to a direct link between the chosen source and destination. Only few packets could be delivered when more than one hop is required. Once the penetration rate increases this trend changes. The ratio of single-hop packets to all transmitted packets only slightly increases to 43% while the remaining communication is provided via multiple hops.

Nonetheless, even with 100% penetration rate the delivery ratio stalls at approx. 90% in the most favorable case. While a failure rate of about 10% may be tolerable by some applications, it is definitely too high for low-latency or mission critical use cases. Additional mechanisms on higher layers to cope with the considerable amount of packet loss are unavoidable.

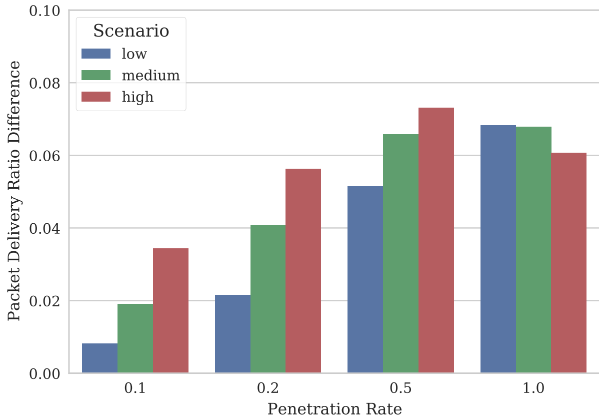


Fig. 3. Packet delivery ratio improvement by applying *select all* instead of *select closer*.

Figure 3 shows the increase in packet delivery ratio if the selection of potential forwarders is not limited to nodes closer to the destination. Similar to the general trend, the benefit increases if more vehicles are available. This can be explained by an increase in forwarding candidates making it more likely to find one in a suitable location. However, the overall increase of at most 7% may explain why more sophisticated strategies to deal with local optima were not included in the ETSI GeoNetworking standard.

Furthermore, the increase in reliability comes at the price of additional hops. In contrast to more realistic approaches the reference algorithm does not suffer from packet loss during a transmission. Nonetheless, the probability of such an error is larger than zero for each hop. Thus, failure probability for the entire path increases with its length. This may counteract the reliability gain by the broader selection policy in a real deployment.

C. Greedy Forwarding

A comparison between the packet delivery ratio achieved by the reference algorithm (*select closer*) and Greedy Forwarding for the same scenario is shown in Fig. 4. There are two observations worth noting: For low penetration rates Greedy performs even slightly better than our reference approach, but for higher network densities it struggles.

The performance advantage of the real protocol over our reference can be explained by the missing randomness in the communication model. While the simulation for Greedy includes a fading component following a Nakagami distribution, the reference algorithm does not. On average fading reduces the received power even further but may increase it occasionally modeling constructive interference of multiple paths. The resulting increase in communication range allows to reach neighbors with a direct link that are discarded by the reference algorithm. However, this effect is only significant for packets delivered via a single hop and diminishes for longer paths.

More vehicles lead to more selection options in each forwarding step. Unfortunately, in Greedy's case more options

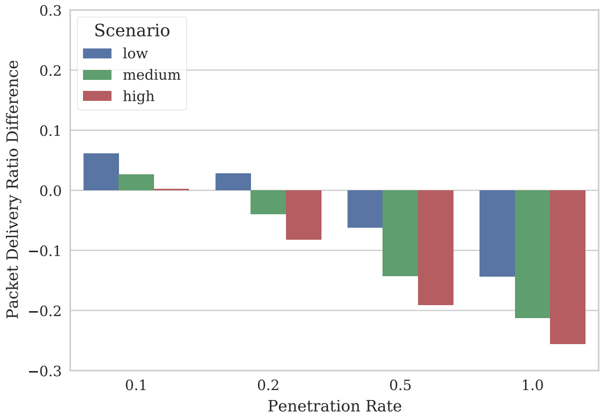


Fig. 4. Packet delivery ratio difference between Greedy Forwarding and the reference algorithm applying the *select closer* strategy.

also lead to more flawed decisions compared to the optimal reference. With increasing network density always selecting the vehicle with the highest distance progress has several drawbacks:

- Optimizing for higher distances between sender and receiver makes it more likely that the selected candidate is not within communication range anymore.
- If the destination is on an adjacent road, nodes in the middle of a road segment have a lower chance of *getting around the corner* than those close to intersections [19].

However, protocols to overcome these drawbacks have already been suggested, e.g. [20], [21], [22], and will be investigated in the future.

V. Conclusion

Vehicular communication is an important step towards cooperative automated driving. In this paper we have investigated properties of multi-hop communication for vehicles in the city of Luxembourg depending on the penetration rate and road traffic density. We introduced an idealistic reference routing algorithm based on a deterministic communication model and compared it to the standardized Greedy approach.

We showed that reliable multi-hop communication depends on a sufficient penetration rate which will not be available from the first day. Thus, augmenting the ad hoc network with other technologies like cellular communication can help to bridge the connectivity cap in the early stages of deployment.

Future work should focus on further improving the communication model with respect to realistic channel behavior. The influence of an increase in transmission range also needs to be investigated. Finally, we intend to apply other existing routing protocols to the same scenario to get more comprehensive results.

Acknowledgment

The research leading to these results has received funding from the Bavarian Ministry of Economic Affairs and Media, Energy and Technology.

References

[1] IEEE, “802.11-2012: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” 2012.
[2] ETSI, “ETSI EN 302 663 V1.2.1 - Intelligent Transport Systems (ITS); Access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency band,” 2013.
[3] A. Tomatis, H. Menouar, and K. Roscher, “Forwarding in VANETs: Geonetworking,” in *Vehicular Ad Hoc Networks Standards, Solutions, and Research*, 2015, pp. 221–251.
[4] ETSI, “ETSI TR 101 607 V1.1.1 - Intelligent Transport Systems (ITS); Cooperative ITS (C-ITS); Release 1,” may 2013.
[5] ——, “ETSI EN 302 637-2 V1.3.2 - Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service,” nov 2014.
[6] K. C. Lee, U. Lee, M. Gerla, L. Uichin, and M. Gerla, “Survey of routing protocols in vehicular ad hoc networks,” *Advances in Vehicular Ad-Hoc Networks: Developments and Challenges, IGI Global*, vol. 21, pp. 149–151, 2009.
[7] B. Karp and H. T. Kung, “GPSR: greedy perimeter stateless routing for wireless networks,” in *Proceedings of the 6th annual international conference on Mobile computing and networking*, ser. MobiCom ’00. New York, NY, USA: ACM, 2000, pp. 243–254.
[8] ETSI, “ETSI EN 302 636-4-1 V1.2.1 - 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 Functionalit,” 2014.
[9] L. Codeca, R. Frank, and T. Engel, “Luxembourg SUMO Traffic (LuST) Scenario: 24 hours of mobility for vehicular networking research,” in *2015 IEEE Vehicular Networking Conference (VNC)*. IEEE, dec 2015, pp. 1–8.
[10] M. Heissenbüttel and T. Braun, “Optimizing neighbor table accuracy of position-based routing algorithms,” in *IEEE INFOCOM*, 2005, p. 1.
[11] L. Cheng, B. E. Henty, D. D. Stancil, F. Bai, and P. Mudalige, “Mobile vehicle-to-vehicle narrow-band channel measurement and characterization of the 5.9 GHz Dedicated Short Range Communication (DSRC) frequency band,” *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 8, pp. 1501–1516, 2007.
[12] C. Sommer, D. Eckhoff, R. German, and F. Dressler, “A computationally inexpensive empirical model of IEEE 802.11p radio shadowing in urban environments,” in *2011 8th International Conference on Wireless On-Demand Network Systems and Services, WONS 2011*. IEEE, 2011.
[13] ETSI, “ETSI TR 102 861 V1.1.1: STDMA recommended parameters and settings for cooperative ITS; Access Layer Part,” 2012.
[14] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, “Recent Development and Applications of SUMO - Simulation of Urban MObility,” *International Journal On Advances in Systems and Measurements*, vol. 5, no. 3&4, pp. 128–138, dec 2012.
[15] “ns-3 (3.25),” 2016. [Online]. Available: http://www.nsnam.org
[16] K. Roscher, J. Jiru, A. A. Gonzalez, and W. A. Heidrich, “ezCar2X: a modular software framework for rapid prototyping of C2X applications,” in *9th ITS European Congress*, Dublin, Ireland, jun 2013.
[17] K. Roscher, S. Bittl, A. A. Gonzalez, M. Myrtus, and J. Jiru, “ezCar2X. Rapid-Prototyping of Communication Technologies and Cooperative ITS Applications on Real Targets and Inside Simulation Environments,” in *Wireless Communication and Information. Digitale Gesellschaft*, 2014.
[18] O. Tange, “GNU Parallel - The Command-Line Power Tool,” *;login: The USENIX Magazine*, pp. 42–47, feb 2011.
[19] C. Lochert, M. Mauve, H. Füßler, H. Hartenstein, H. Füssler, and H. Hartenstein, “Geographic routing in city scenarios,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 9, no. 1, pp. 69–72, jan 2005.
[20] M. Heissenbüttel, T. Braun, M. Heissenb, and T. Braun, “BLR: Beacon-Less Routing Algorithm for Mobile Ad-Hoc Networks,” *Elsevier’s Computer Communications Journal*, vol. 27, pp. 1076–1086, 2003.
[21] K. C. Lee, U. Lee, and M. Gerla, “TO-GO: TOpology-assist Geo-Opportunistic Routing in Urban Vehicular Grids,” in *WONS 2009 - 6th International Conference on Wireless On-demand Network Systems and Services*. IEEE, 2009, pp. 11–18.
[22] K. Roscher, J. Jiru, and R. Knorr, “Low-delay forwarding with multiple candidates for VANETs using multi-criteria decision making,” in *2016 IEEE Vehicular Networking Conference (VNC)*. IEEE, dec 2016, pp. 1–8.

Proceedings of the 5th GI/ITG KuVS Fachgespräch Inter-Vehicle Communication (FG-IVC 2017)
Anatoli Djanatliev, Kai-Steffen Hielscher, Christoph Sommer, David Eckhoff, and Reinhard German (Eds.)

Communication Characteristics of VANETs

Christina Stadler*, Xenia Flamm*, Thomas Gruber*, Reinhard German† and David Eckhoff‡
* AUDI AG, Ingolstadt, Germany
† Computer Networks and Communication Systems, Dept. of Computer Science, University of Erlangen, Germany
‡ TUMCREATE, Singapore
{christina1.stadler,xenia.flamm,gruber.thomas}@audi.de
reinhard.german@fau.de, david.eckhoff@tum-create.edu.sg

***Abstract*—The standardization work for Vehicle-to-Vehicle (V2V) communication based on IEEE 802.11p has advanced enough for the start of application development. Future providers of V2V communication applications need to know the technical boundaries of their products to be able to keep their promises towards the customers. As these applications are based on the information received from other road users, their successful operation depends on the communication quality provided by the surrounding setting. With many obstacles such as buildings, foliage and infrastructure, urban scenarios pose a challenge to the communication quality for Vehicular Ad-Hoc Networks (VANETs). This paper investigates the communication characteristics of different urban environments in terms of Line-of-Sight probability. We find a strong correlation between the chance of a building blocking the line of sight not only based on the distance but also on the angle between two vehicles. The latter property has not yet received wide attention from the research community, however, we believe it opens interesting ways both to design future applications and to stochastically model urban V2V communication.**

I. Introduction

Future services and applications based on V2V communication require a certain amount of information in a certain time and/or over a certain spatial expansion to provide their envisioned quality of service. If these requirements are not met and the promised effect cannot be reached, the comfort or even the safety of a driver suffers. In order to prevent the client from disenchantment and the provider from an image loss, technical limitations need to be taken into account when developing new applications. These requirements depend on the communication quality available in the field. For that purpose, this paper introduces preliminary investigations to automatically estimate the communication characteristics of a specific urban scenario.

The surrounding environment of a VANET determines the communication characteristics and thus defines the communication quality. In this paper we introduce and examine the hypothesis, that urban scenarios do have a deterministic communication characteristics based on their topology. In the following, these characteristics will be represented by the probability for a Line-of-Sight (LOS) link between two communication partners depending on their distance and their angle to each other. We found out that the angle between vehicles is a crucial parameter for the differentiation between scenario topologies' influence on VANETs. In every situation a object leads to a Non-Line-of-Sight (NLOS) condition between communication partners, it potentially attenuates the radio signal and thereby negatively affects the communication quality. Hence a LOS condition will always provide better communication quality than NLOS. Karedal [1] for example shows with measurements that the absence of LOS is a severe problem for collision avoidance applications. Focussing solely on LOS probabilities, we can isolate effects caused by the topology of a scenario, as this probability is not dependent on communication parameters such as the transmission power.

II. Related work

The effect of urban scenarios on vehicular communication has been studied by both industry and academia. Oishi et al. examined the influence of the building density within a scenario on channel characteristics [2]. The paper uses the building density to describe the influence of the inter-vehicular angle on the LOS probability between communication partners. For that purpose, it considers an artificial road network with solely 90 degree angles and rectangular buildings. This simplified synthetic scenario, however, cannot be transferred to real urban scenarios that have different topologies.

Samimi et al. [3] and Sun et al. [4] define a model describing the influence of an urban scenario on millimeter-wave outdoor communication and 5G communication, respectively. In contrast to Oishi [2], Samimi [3] and Sun [4] conducted their investigations with real urban scenarios, but they did not examine the difference in the impact on the communication of various urban scenarios and the important role of the inter-vehicular angle.

In this paper, we want to give a more realistic and a broader view on the communication characteristics of Inter-Vehicle Communication (IVC) depending on the setting of urban scenarios. We do that by investigating various typical categories of real urban scenarios in Germany.

III. Evaluation Parameters

Our hypothesis is that different categories of urban scenario settings have different LOS probabilities. Additionally, we suppose that this relation is deterministic and can be modeled.

To examine this hypothesis, both the influencing factors, that is, the parameters of a scenario, as well as resulting communication characteristics, need to be evaluated. These are introduced in the following two sections.

(a) Historical Center of Ingolstadt, Germany (HistCtr)
(b) Feselenstraße, Ingolstadt, Germany (Urban1)
(c) Richard-Wagner-Straße, Ingolstadt, Germany (Urban2)

Figure 1. Three scenarios of 800 m x 800 m, buildings are colored in red.

A. Scenario Attributes

For our investigations we selected urban scenarios from four different categories: historical center, rural residential area, urban residential area and industrial area. Concerning the topology, we tried to capture the highest variance between the urban settings and considered preferably homogeneous scenarios to avoid ambiguous results.

LOS probability values are determined by the road network and the arrangement and extension of buildings. The roads define possible geometrical setups of VANETs in relation to the buildings within a scenario. Hence, the geometrical properties of buildings and roads represent the scenario attributes decisive for the communication characteristics. For the sake of comparability, these attributes need to be measurable and extractable, e.g., from maps described in XML.

The road network attributes are given in the following list:

1) average length of road segment in [m]
2) number of roads
3) percentage area covered by roads
4) number of curves
5) number of intersections

A road consists of several lane segments and ends as soon as there is no consecutive lane that forms an angle between the next and the current lane segment, that lies in the range of 20° and 160°. This definition is based on the assumption that no building along a road can intersect a communication link of two vehicles. The number of curves is indicated by the number of angles between two roads within the range of 20° and 160°. An intersection is any junction of two roads with an angle between 45° and 135°. These definitions should be perceived as indicators to capture the differences in various urban scenarios.

Concerning the buildings in a given scenario, we consider the following list of attributes:

6) average building area in [m^2]
7) number of buildings
8) percentage area covered by buildings

The attributes in Table I show the distinction in topology between the three scenarios depicted in Figure 1. The scenario attribute values of the two urban residential scenarios are similar and differ significantly from the value for the historical center. In contrast to the urban residential areas, the historical center is characterized by short yet many roads with a large number of curves and intersections. Although the buildings within the historical center are smaller, the amount of the buildings is so high that the percentage area covered by buildings is twice as high as in the urban residential area.

Table I
SCENARIO ATTRIBUTES OF THE HISTORICAL CENTER AND TWO URBAN RESIDENTIAL AREAS IN INGOLSTADT, GERMANY

Attributes	**HistCtr**	**Urban1**	**Urban2**
average length of road segment	87 m	143 m	104 m
number of roads	127	67	79
perc. area covered by roads	10.4%	9.2%	7.9%
number of curves	258	162	159
number of intersections	1117	57	66
average building area	304 m^2	400 m^2	567 m^2
number of buildings	1035	413	251
perc. area covered by buildings	46.4%	24.7%	21.3%

It is important to state that this list is not complete as it does not, for example, capture the shapes and constellations of the given buildings. The consideration of more detailed influence factors is left to future work.

B. Parameters of Communication Characteristics

This paper focuses on the impact of the geometrical scenario setting on VANETs characteristics. Thus this influence factor needs to be isolated from other influencing factors like the material a radio wave penetrates through (i.e. the facade of buildings or the air humidity). The focus on the geometry of a scenario happens by the evaluation of the chance to meet a LOS constellation, which offers communication conditions far better than NLOS constellations. The presence of a LOS gets evaluated between a sender and all potential receivers for every sent message. As introduced in Section I, the LOS probability serves as an indicator for the communication quality expectable for the examined link. The distance and angle between possible communication partners are decisive parameters for the geometrical description of VANETs. Hence the LOS probability was investigated regarding those two parameters. The results are discussed in the following section.

IV. COMMUNICATION CHARACTERISTICS

To examine the LOS probability of urban scenarios we used the Veins framework [5]. Simulations of several scenarios were done with the same simulation setup. Periodic beacons were sent with a constant rate of 10 Hz, a transmission power of 20 mW and a receiver sensitivity of -89 dBm. The size of each simulated scenario was 800 m x 800 m with a VANET of 100 vehicles in it. The vehicles were randomly placed on routes, which were automatically generated weighted by the length and the number of lanes of a street. We conducted extensive simulation runs for each selected scenario and also computed the scenario parameters presented in Subsection III-A. The output of one simulation run consisted of about 5 Mio. data sets, one for every simulated communication link. For each

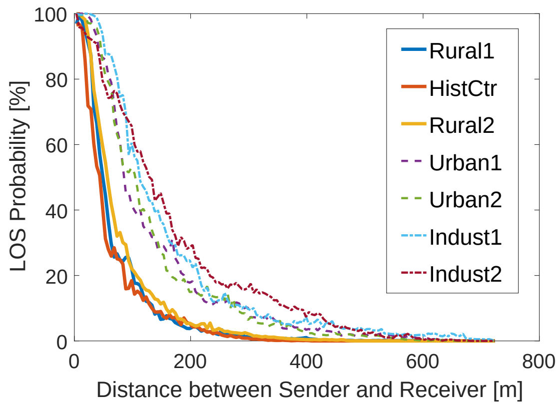


Figure 2. LOS probability over distance between sender and receiver for seven scenarios in and around Ingolstadt, Germany.

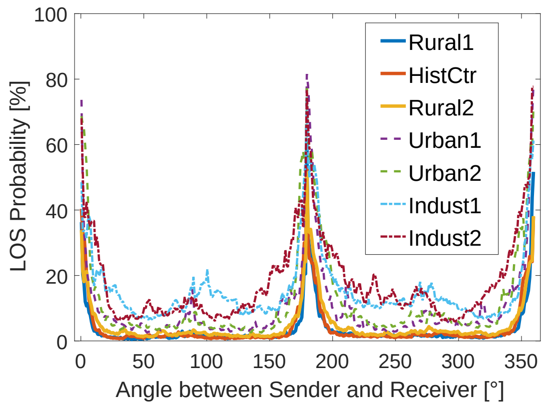


Figure 3. LOS probability over angle between sender and receiver for seven scenarios in and around Ingolstadt, Germany.

link the following information was collected: sender-receiver distance, angle between sender and receiver, LOS/NLOS.

The evaluation of the communication characteristics of VANETs depending on the surrounding scenario topology was done in three steps. For randomly distributed vehicles in an urban scenario the LOS probability decreases with increasing distance between sender and receiver. The larger the distance between two communication partners, the higher the probability for the presence of a building intersecting the communication link. This obvious relation will be investigated in the first step. Motivated by the findings presented in [6], the influence of the inter-vehicular angle on the LOS probability will be discussed secondly. Lastly, the combined impact of distance and angle is evaluated under consideration of the scenario-specific attributes.

A. LOS Probability over Distance

Our assumption is that urban scenarios with similar topologies have similar communication characteristics and vice versa. To examine the similarities and differences of the simulation results regarding the topology, scenario attributes were introduced in Subsection III-A. Table I contrasts these attributes for the historical center and the two urban residential scenarios in Ingolstadt, Germany. Figure 2 shows the simulation results for seven scenarios in and around Ingolstadt, Germany, evaluated with respect to the LOS probability over the sender-receiver distance.

At first glance there is an obvious similarity within the urban residential areas' attribute values and an obvious distance to the historical areas' values. Figure 2 shows the same behavior concerning the LOS probability, supporting our hypothesis. In comparison to the historical scenario the urban residential scenarios contain long roads, with few crossings and curves. That indicates a low level of entanglement within the road network. Due to long, straight roads without buildings intersecting a communication link, the LOS probability remains high over large distances for urban residential areas. In contrast to that, a large number of intersections and short roads decrease the LOS probability already for smaller distances in the historical center.

The average building area of a scenario and the entanglement of a road network interact, as large buildings determine long, straight roads. That relation is also observable in Table I for the urban residential scenarios. The other case, where smaller buildings lead to a more angular road network, is shown by the historical scenario. As expected, a higher building density induces an overall lower chance for LOS conditions.

Comparing all four scenario categories, Figure 2 provides the following findings: There are two obvious groups of curves. One group contains the rural residential areas and the historical center showing the same behavior. The second group consists of the urban residential and the industrial areas with a small difference between them, but not as large as to the first group. The latter provides the worst communication conditions, as the decrease of their LOS probability starts for smaller distances and proceeds with a higher gradient. The industrial areas tend to provide the best communication conditions as their LOS probability decreases with the smallest gradient over the distance.

B. LOS Probability over Angle

We introduced the entanglement as a property of road networks described by the combination of the following scenario attributes: Average road length, number of curves, and crossings. To evaluate the influence from road entanglement on the V2V communication, we analyzed the LOS probability over the angle between sender and receiver. For that purpose, we used the same angle definition as [6].

[6] describes the results of a simulation-based examination of the impact of different antenna patterns on VANETs. As mentioned before, this paper focuses on the impact of urban settings, hence it uses an idealistic isotropic antenna. In future work different antenna patterns can easily be integrated. Figure 3 shows similar results for the LOS probability over the angle as [6] does for the the number of received packets. That additionally confirms the LOS probability as a suitable

topology-based indicator for the communication characteristics of VANETs.

Figure 3 visualizes the LOS probability over the inter-vehicular angle for the same seven scenarios as Figure 2 does over the distance. Vehicles driving on the same road form angles of about 0°/360° or 180° to each other. The simulation results verify the assumption that vehicles on the same road have the significantly highest LOS probabilities within an urban scenario. The LOS probability at the angle value 180° for the urban residential scenarios are nearly 80%. Whereas the historical center provides only a 50% LOS probability. As a result, the possibility for vehicles driving the same road is higher in urban residential areas than in an historical area. That fact is also provable by the scenario attributes in Table I. Urban residential areas contain a lower level of entanglement by a larger average road length and fewer curves and intersections than the historical center. Additionally, the smaller percentage of area covered by buildings in urban residential areas facilitates higher LOS probabilities for vehicles not using the same road. These cases are represented by the angle values other than 0°, 180°, and 360°. All in all, the three groups of lines that are observable in Figure 2 are also visible in Figure 3 and they are arranged in the same order.

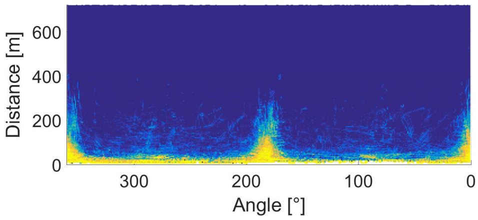
C. LOS Probability over Distance and Angle

In this section we examine the distance and angle between sender and receiver as common predictors for the LOS probability. The evaluation results, represented in Figure 4, show that there is a higher LOS probability for larger distances around the angles of 0°/360° and 180°. This points out the importance of the inter-vehicular angle. The Figures 4b and 4c demonstrate the similarity of two scenarios with a similar topology. A comparison of these two heatmaps with the one in Figure 4a shows clear differences. This supports the assumption of a site-specific influence on VANETs. In a following step, a representative cross-section of German cities, about thirty urban scenarios were evaluated, large cities, as well as small cities. The same observations were made.

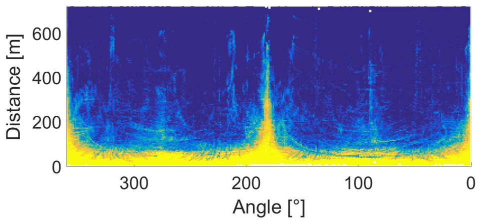
These investigations led to the following assumption: Today, due to neighbor and fire protection reasons, the arrangement of buildings is regulated by law. Areas like historical centers, that evolved over centuries, contain a high level of entanglement as they grew slowly and did not have to follow any regulations regarding the placement and dimension of buildings. However, new urban residential areas, that are artificially designed as a whole, do have to observe the law and therefore contain a low level of entanglement. In rural areas a mixture of both grown historical center and new artificial areas is observable.

V. Conclusion and Future Work

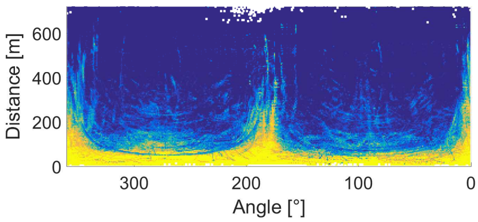
In this paper, the impact of an urban scenario topology on the communication characteristics of VANETs is presented. This is done by the evaluation of the influence that distance and angle between sender and receiver vehicles have on the LOS probability of a communication link. Furthermore, this paper proposes a measure by introducing site-specific attributes to



(a) Historical center Ingolstadt, Germany (HistCtr)



(b) Feselenstraße, Ingolstadt, Germany (Urban1)



(c) Richard-Wagner-Straße, Ingolstadt, Germany (Urban2)

Figure 4. Heatmaps for three German scenarios with the angle between vehicles on the x axis and the distance on the y axis. Lighter colors represent a higher LOS probability.

calculate road and building characteristics of a given scenario. This can be transferred to new areas.

In a future step we will elaborate a model that predicts the influence of a scenario topology on the communication characteristics of a VANET by the scenario attributes for every urban scenario. For that purpose, the sufficiency of the introduced scenario attributes needs to be evaluated.

References

[1] J. Karedal, F. Tufvesson, T. Abbas, O. Klemp, A. Paier, L. Bernado, and A. F. Molisch, "Radio channel measurements at street intersections for vehicle-to-vehicle safety applications," in *71st IEEE Vehicular Technology Conference (VTC 2010-Spring)*. IEEE, 2010.

[2] J. Oishi, K. Asakura, and T. Watanabe, "A Communication Model for Inter-vehicle Communication Simulation Systems Based on Properties of Urban Areas ," *IJCSNS International Journal of Computer Science and Network Security*, vol. 6, no. 10, pp. 213–219, October 2006.

[3] M. K. Samimi, T. S. Rappaport, and G. R. MacCartney, "Probabilistic Omnidirectional Path Loss Models for Millimeter-wave Outdoor Communications," *IEEE Wireless Communications Letters*, vol. 4, no. 4, pp. 357–360, August 2015.

[4] S. Sun, T. A. Thomas, T. S. Rappaport, H. Nguyen, I. Z. Kovacs, and I. Rodriguez, "Path loss, shadow fading, and line-of-sight probability models for 5g urban macro-cellular scenarios," in *Global Communications Conference Workshop (Globecom Workshop)*. San Diego, CA, USA: IEEE, December 2015.

[5] 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.

[6] D. Eckhoff, A. Brummer, and C. Sommer, "On the impact of antenna patterns on VANET simulation," in *Vehicular Networking Conference (VNC)*. IEEE, 2016.

Proceedings of the 5th GI/ITG KuVS Fachgespräch Inter-Vehicle Communication (FG-IVC 2017)
Anatoli Djanatliev, Kai-Steffen Hielscher, Christoph Sommer, David Eckhoff, and Reinhard German (Eds.)

Vanetza: Boosting Research on Inter-Vehicle Communication

Raphael Riebl, Christina Obermaier, Stefan Neumeier, Christian Facchi
Research Centre, Technische Hochschule Ingolstadt
Email: {raphael.riebl, christina.obermaier, stefan.neumeier, christian.facchi}@thi.de

***Abstract*—A plethora of applications are envisaged based on wireless communication between vehicles, promising novel Advanced Driver Assistance Systems for safety, comfort and efficiency. Prototyping, testing and evaluation of such applications is challenging when no appropriate tools are readily available. Vanetza is our contribution towards a comprehensive open-source environment for Vehicle-to-Anything (V2X) communication as specified by the ITS-G5 standards for Europe. A brief comparison with other currently available ITS-G5 software highlights Vanetza's uniqueness in the field of Vehicular Ad Hoc Network (VANET) communication stacks. Using Vanetza as ITS-G5 implementation for research and testing is a natural fit because it has been explicitly designed for these purposes. This paper outlines the features currently supported by Vanetza and how it can be deployed for diverse use cases ranging from embedded systems to large-scale simulations. An outlook of future development directions finishes this momentary insight into Vanetza off.**

***Index Terms*—Inter-Vehicle Communication, GeoNetworking, ETSI ITS-G5, Vehicular Ad Hoc Network**

I. Introduction

Without a doubt, Vehicle-to-Anything (V2X) communication will change mobility in the foreseeable future. The European Commission reaffirmed its support for Cooperative Intelligent Transport Systems (C-ITS) recently and named the hybrid combination of ETSI ITS-G5 and cellular networks the most promising V2X communication mix [1]. Likewise, the US legislature is pushing V2X technology through a rulemaking process [2]. Both sides of the Atlantic build their hopes on V2X communication for more traffic safety by reducing accidents and their ramifications as well as increased mobility efficiency by reducing traffic congestions.

In Europe, V2X communication is standardised by ETSI as an architecture named ITS-G5 [3]. This paper highlights Vanetza, an open-source implementation of several ITS-G5 standards. While other implementations also exist, which are looked at in Section II, Vanetza is nonetheless unique as it is the only freely available ITS-G5 protocol stack currently. In Section III, standardised as well as experimental key features of Vanetza are presented. Vanetza's versatility is then outlined in Section IV by reference to several usage scenarios, e.g. simulations, embedded systems and testbeds. Last but not least, gaps for future development and third-party contributions are named in Section V.

II. Related V2X Software

The majority of available V2X communication stacks is not freely available but tightly bound to particular products. Even if the acquisition of such a stack may be affordable, it remains hard to use these stacks in another way than the one foreseen by the original vendor. Many communication stacks are designed for a particular hardware environment and may only work when this hardware is present.

Proprietary V2X communication stacks hence suffer from at least one or more of following problems which restrict their usefulness for research:

- *Blackbox behaviour:* Lacking source code availability the behaviour of a V2X communication stack can only be observed at its API and radio interface. Despite a test suite provided by ETSI, this hinders a thorough analysis and validation of the implementation.
- *Customisability:* The degree of customisability depends largely on the provided API. More radical customisations tend to be increasingly difficult if not impossible.
- *Sharing:* Modifications and extensions are difficult to share with the research community because the underlying proprietary implementation cannot be shared.
- *Costs:* Commercial stacks are more or less expensive. This might be obstructive for new researchers to the field.

Table I lists available V2X software applicable for the ETSI ITS-G5 communication. For sure, this list is by no means exhaustive but represents those options we know about through publicly available information. We do not advertise any particular solution but recommend to compare those carefully if they fulfil individual requirements. Some of the commercial solutions might not be totally unique implementations at all but contain licensed third-party code. Autotalks, a developer of V2X chipsets, does not offer a own software stack but claims that eight third-party implementations are available for their chipset CRATON2[1].

The C2X SDK [4] previously offered by NEC Laboratories Europe for research purposes, is no longer available as of January 2017[2]. Consequently, it boils down to three open-source implementations dealing with ITS-G5: Voronov *et al.* share a Java library under the terms of Apache License 2.0. Their library can be used to generate packets conveying Cooperative Awareness Messages (CAMs) or Decentralized Environmental Notification Messages (DENMs) [5]. This library provides neither packet routing nor an ITS-G5 security

[1] http://www.auto-talks.com/wp-content/uploads/2016/10/SOFTAWARE.pdf (Accessed on 21.03.2017)

[2] http://c2x-sdk.neclab.eu (Accessed on 17.03.2017)

Table I
SUMMARY OF ITS-G5 SOFTWARE

Name	License	Short description and reference
Cohda Wireless SDK	Proprietary	Requires a Cohda chipset and Cohda's extensions to the Linux kernel http://cohdawireless.com/Portals/0/MKx_SDK_10122015.pdf
Commsignia	Proprietary	Full stack running on Linux and automotive real-time OS http://www.commsignia.com/software/
ezCar2x by Fraunhofer ESK	Proprietary	Collection of C++ libraries intended for prototyping and simulations https://www.esk.fraunhofer.de/en/research/projects/ezCar2X.html
Kapsch TrafficCom	Proprietary	Included in their systems (EVK-3300, TS3306 and others), seems to run on Linux https://www.kapsch.net/ktc/downloads
Marben V2X	Proprietary	Full stack, C++ source code available, claims to be highly portable (OS and hardware) http://marben-products.com/v2x/v2x-software-solution.html
NORDSYS waveBEE	Proprietary	Runs on Linux-based waveBEE boxes intended for application development http://nordsys.de/en/car2x-produkte-2/280-wavebeesoftware.html
OpenC2X	Open-source (LGPLv3)	Focussed on applications, no support of network protocols http://www.ccs-labs.org/software/openc2x/
Vanetza	Open-source (LGPLv3)	C++ library covering GeoNet, BTP, DCC and Security (Facilities only rudimentary) http://www.vanetza.org
Voronov's et al. GeoNetworking	Open-source (Apache 2.0)	Java implementation (incomplete for Day One usage, e.g. missing packet forwarding) https://github.com/alexvoronov/geonetworking

layer implementation, though, which are required features for Day One ITS stations. Researchers preferring the Java ecosystem might still find it useful as their starting point.

OpenC2X by Laux *et al.* aims to establish a prototyping platform for ITS-G5 [6]. Using off-the-shelf hardware one can deploy a home-brew V2X communication device. Version 1.0 of OpenC2X includes some effort to gather required data through a vehicle's OBD-II interface and a GPS receiver. Otherwise it seems to focus on functional aspects of V2X applications but not on a fully operational implementation of GeoNetworking or the ITS-G5 security layer.

Vanetza, then again, aims at providing generic ITS-G5 networking features but is not a standalone software by design. Programs linking to the Vanetza libraries are expected to take care of platform specifics, e.g. binding Vanetza to a wireless radio interface. Generic features such as GeoNetworking, Basic Transport Protocol (BTP) and Decentralized Congestion Control (DCC) are rather unaffected by the hosting V2X platform. Retrieval of vehicle parameters such as position or speed, though, depends highly on the target platform. Thus, integration into a specific vehicular environment has to be realised by the linking program.

III. FEATURES

ITS-G5 components, which are mostly independent of the intended target platform, are in the focus of Vanetza. This comprises the network and transport layers as well as the congestion control and security cross-layers, which are coloured green in Figure 1.

Vanetza is written in C++11 and depends on a few portable libraries, i.e. it can be built for almost any system where a standard-compliant C++11 compiler and these dependencies (Boost, GeographicLib, Crypto++ and optionally OpenSSL) are available. Platforms known to work so far are Linux (x86, ARM) and Windows (x86).

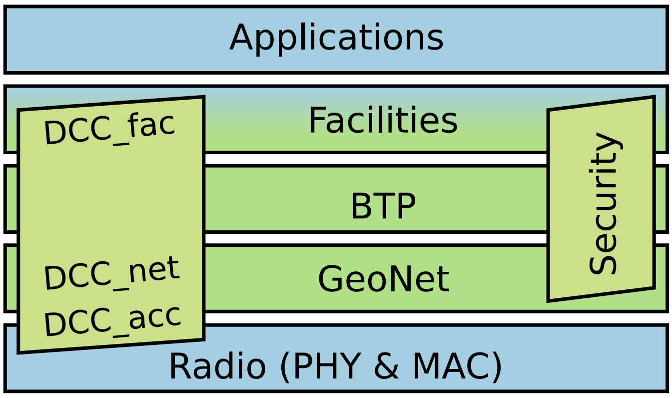


Figure 1. Layers supported by Vanetza (green) in the ITS-G5 architecture

A. Basic principles

There are several components in ITS-G5 depending on time progress, e.g. expiry of location table entries, packet routing or repetitions. Vanetza, however, is a purely reactive system on purpose, i.e. no Vanetza code runs without stimuli from outside. Those stimuli can be incoming and outgoing packets but also time triggers. Such time triggers are supported by Vanetza through its *Runtime* instance, which can be used by other components for scheduling timed callbacks. Progress of *Runtime* in turn is controlled from outside, e.g. a system clock read by the enclosing program.

Packets are a substantial data structure in any network protocol stack. When an outgoing packet traverses the Vanetza protocol stack, each layer can add its header information in form of data structures. Ownership of a packet is passed on layer by layer so copying costs are reduced. For over-the-air transmissions those structured packets can be serialised to plain bytes on request. Vice versa, layer boundaries in incoming packet byte streams can be marked. This eases the parsing and extracting of protocol headers.

B. GeoNetworking

GeoNetworking, the network layer in ITS-G5, provides many features, e.g. routing of packets based on geographic positions and areas, detection of duplicate packets, store & carry forwarding for delay-tolerant transmissions, and packet repetitions [7]. Albeit only the feature subset required for Day One deployment is implemented, *geonet::Router* is one of Vanetza's most powerful classes. Most notably, the Single Hop Broadcast (SHB) and GeoBroadcast (GBC) modes required for CAM and DENM dissemination are included.

C. Security

Encapsulation of packets into a security envelope is an optional ITS-G5 feature that can be controlled for each *geonet::Router* individually. If activated, all data requests from upper layers are automatically put in such an envelope, i.e. the upper layer's payload is cryptographically signed. When a signed packet is received, the signature and the corresponding certificate are checked for validity. While processing and creating secured packets is an in-built component, Vanetza relies on third-party libraries like Crypto++ or OpenSSL for the cryptographic calculations.

D. Congestion Control

Similar to Security, DCC is a cross-layer in ITS-G5, i.e. its sub-entities interact with other layers as shown in Figure 1. *DCC_acc* acts as gatekeeper above the access layer, i.e. it enforces minimum time intervals between outgoing packets of each priority class. *DCC_fac* adapts the message generation rate, so no packets are delayed or dropped by *DCC_acc*. *DCC_net* – whose implementation is currently work in progress – is supposed to share Channel Busy Ratio (CBR) measurements among neighbouring nodes.

E. Experimental Extensions

Due to its open nature, Vanetza is predestined for experimental features. These can be variants of existing, standardised features or completely new ideas. Some non-standard extensions are already included and can be enabled on demand. One example concerns the *Security* layer where whole backend implementations can be exchanged. It is also the *Security* entity which allows for customisation regarding the time point of actual signature calculation, so-called "deferred signing" [8].

Further adjustments can be made at DCC: While standardisation dictates a state machine for evaluating channel congestion, this can alternatively be done by a linear function. Last but not least, the behaviour and length of DCC queues is not fixed at all and allows for interesting experiments with respect to buffer bloat phenomena.

IV. Use Cases

Vanetza has been designed and implemented with portability and flexibility in mind. Thus, Vanetza is not eagerly optimised for a single use case but can be used in several scenarios. A selection of those use cases and how they integrate Vanetza are presented hereinafter.

A. VANET Simulations

Simulations of hundreds to thousands vehicles interconnected by a Vehicular Ad Hoc Network (VANET) have been the very first purpose of Vanetza. Since the beginning of this simulation as tiny extension [9] of Veins it has grown to the sophisticated V2X simulation framework "Artery" [10]. Meanwhile, Artery incorporates INET as alternative radio model and supports LTE cellular communication [11] as well as ranging sensors attached to vehicles for environmental perception [12], [13]. Despite the growing set of features, the integration of Vanetza into Artery still touches three main aspects:

a) time progress: Simulation time in discrete event simulations is determined by the time stamp of the last processed event. When simulating a huge number of vehicles, the sheer number of events and their corresponding execution time forbids hard synchronisation between simulation and wall-clock time. Thus, the simulation time is used for triggering Vanetza's *Runtime* instead of a system clock.

b) packet handling: For performance reasons, packets are not converted into byte buffers in simulations, i.e. the serialisation step at transmitter and deserialisation steps at each receiver are omitted. Instead, receiving network nodes can read the data structures directly, which have been assigned by the transmitting node before.

c) interfaces to neighbouring layers: Each vehicle is equipped with a simulated radio model comprising the physical and access layer for wireless communication. These layers are either provided by Veins or INET. Either way, Artery provides small adaptors fulfilling Vanetza's *access::Interface* for transmitting packets. Received packets are passed as data indications to each vehicle's *geonet::Router*. With respect to the application layer, vehicles process CAMs and DENMs by use of Artery services.

Though Artery builds upon OMNeT++, there are no technical reasons hindering the integration of Vanetza into other discrete event simulations such as ns-3.

B. Embedded Systems

Vanetza itself does not require any operating system specific Application Programming Interface (API). Thus, any reasonable powerful embedded systems can use Vanetza as protocol stack. At least, cross-compiled versions of Vanetza's dependency libraries and a C++11 capable compiler are required.

Prototypically, Vanetza has been deployed on the *MKx* platform from Cohda Wireless. Among others, this deployment exerts the proprietary *MKx* API for setting transmission power, modulation coding scheme and priority class per packet. Though this platform comes with its own, proprietary ITS-G5 stack, running Vanetza on it is still beneficial: While the over-the-air packets can be inspected using network sniffers such as Wireshark, the stack processing itself is hidden in Cohda Wireless' kernel modules. Vanetza can be used as a whitebox reference implementation on this platform.

C. Continuous Integration

Beside VANET simulations and embedded systems, Vanetza can also be deployed in automated test environments. Ideally, test cases consume only little time so developers do not refrain from running them often. Slow test execution might even stall the development process when a continuous integration system has to approve new commits.

V2X application tests should not neglect network behaviour entirely for a speed-up, though. Otherwise test quality might suffer especially for applications relying on multi-hop message dissemination. For example, DENMs are disseminated using GeoBroadcast, a routing paradigm using other stations as forwarders. If no precise timing of channel access or extreme channel conditions shall be tested, a detailed model of radio propagation effects and network interface card behaviour can be neglected. Compared to more detailed VANET simulations the simplified test case execution picks up speed considerably.

A possible realisation might employ a simple directed graph representing the connectivity between network nodes. This could look similar to Vanetza's routing unit tests employing *NetworkTopology*, where each *geonet::Router* is assigned a set of other reachable router instances. With this setup, network and connectivity effects are still covered by the test environment with minimum overhead. Furthermore, these tests can be executed without any special hardware in contrast to a testbed incorporating devices-under-test, i.e. actual V2X hardware.

D. Education

Network stacks are usually an integral part of the operating system and thus not easily accessible for educational purposes. Experimenting and training with V2X networks is alleviated when network functions can be debugged and investigated in user-space with common tools also used for traditional application development. Of course, source code availability is another bonus in this case.

In the course of an one-semester class, students have gained hands-on experience by developing a show-case application based on Vanetza. This application – available as "socktap" tool in the Vanetza repository[3] – enables deployment of the ITS-G5 architecture on commodity hardware. Students have used the Raspberry Pi with its integrated 802.11 Wireless LAN module and attached GPS receiver as target platform in this particular case. Wireless communication has been realised through Linux packet sockets, i.e. upper layers starting with the network layer are running as user-space program. Due to hardware and driver limitations, however, this application is bound to conventional communication in the 2.4 GHz band and lacks channel congestion measurements.

V. Conclusion & Future

Since the beginnings of Vanetza in 2013 it has evolved to a powerful implementation of the ETSI ITS-G5 stack. Successful deployments in simulations as well as on V2X devices have demonstrated its readiness for Day One use cases. Except for Vanetza, there is no other open software currently available to close the gap between application and access layer.

Yet, Vanetza must not be considered finished and further features are going to be implemented as the V2X industry moves on to Day Two and beyond. There are ideas to extend Vanetza into various directions: While secured packets are already supported, an integration with Public Key Infrastructure (PKI) is missing yet. Future applications might also require further network modes beside SHB and GBC, e.g. GeoUnicast (GUC). In that case, the GeoNetworking location service is a prerequisite. If road side units were available over a wide area, then "IPv6 over GeoNetworking" might get more attention too. Contributions to these and other areas are welcomed, even if they are experimental.

[3]https://github.com/riebl/vanetza

References

[1] European Commission, *COM(2016) 766: A European strategy on Cooperative Intelligent Transport Systems, a milestone towards cooperative, connected and automated mobility*, Brussels, Nov. 2016.

[2] National Highway Traffic Safety Administration, Department of Transportation, *NHTSA-2016-0126: Federal Motor Vehicle Safety Standards, V2V Communications*, Jan. 2017.

[3] ETSI, *EN 302 665 - Intelligent Transport Systems (ITS); Communications Architecture*, V1.1.1, European Telecommunications Standards Institute, Sep. 2010.

[4] A. Festag, R. Baldessari, W. Zhang, and L. Le, "CAR-2-X Communication SDK - A Software Toolkit for Rapid Application Development and Experimentations", in *2009 IEEE International Conference on Communications Workshops*, Jun. 2009, pp. 1–5. DOI: 10.1109/ICCW.2009.5208070.

[5] A. Voronov, J. De Jongh, D. Heuven, and A. Severinson, *Implementation of ETSI ITS G5 GeoNetworking stack, in Java: CAM-DENM / ASN.1 PER / BTP / GeoNetworking*, Jun. 2016. DOI: 10.5281/zenodo.55650.

[6] 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 *8th IEEE Vehicular Networking Conference (VNC 2016), Demo Session*, Columbus, OH: IEEE, Dec. 2016, pp. 152–153. DOI: 10.1109/VNC.2016.7835955.

[7] ETSI, *EN 302 636-4-1 - 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*, V1.2.1, European Telecommunications Standards Institute, Jul. 2014.

[8] R. Riebl, M. Monz, S. Varga, L. Maglaras, H. Janicke, A. H. Al-Bayatti, and C. Facchi, "Improved Security Performance for VANET Simulations", in *4th IFAC Symposium on Telematics Applications (IFAC-PapersOnLine)*, vol. 49, Porto Alegre, Brasil: Elsevier, 2016, pp. 233–238. DOI: 10.1016/j.ifacol.2016.11.173.

[9] R. Riebl and C. Facchi, "Implementation of Day One ITS-G5 Systems for Testing Purposes", in *Proceedings of the 2nd GI/ITG KuVS Fachgespräch Inter-Vehicle Communication (FG-IVC 2014)*, 2014, pp. 33–36.

[10] R. Riebl, H.-J. Günther, C. Facchi, and L. Wolf, "Artery: Extending Veins for VANET applications", in *2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, Jun. 2015, pp. 450–456. DOI: 10.1109/MTITS.2015.7223293.

[11] J. Timpner, M. Wegner, H.-J. Günther, and L. C. Wolf, "High-Resolution Vehicle Telemetry via Heterogeneous IVC", in *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc): IoV-VoI Workshop*, Paderborn, Germany: ACM Press, 2016, pp. 19–24. DOI: 10.1145/2938681.2938687.

[12] H.-J. Günther, O. Trauer, and L. Wolf, "The Potential of Collective Perception in Vehicular Ad-hoc Networks", in *2015 14th International Conference on ITS Telecommunications (ITST)*, Dec. 2015, pp. 1–5. DOI: 10.1109/ITST.2015.7377190.

[13] H.-J. Günther, R. Riebl, L. Wolf, and C. Facchi, "Collective Perception and Decentralized Congestion Control in Vehicular Ad-hoc Networks", in *2016 IEEE Vehicular Networking Conference (VNC)*, Dec. 2016, pp. 1–8. DOI: 10.1109/VNC.2016.7835931.

Providentia

Proactive Video-based Use of Telecommunications Technologies in Innovative Motorway Scenarios

Gereon Hinz
fortiss GmbH
An-Institut of Technische Universität München
Germany

Martin Büchel
fortiss GmbH
An-Institut of Technische Universität München
Germany

Josef Eichinger
Huawei Technologies Duesseldorf GmbH
Munich Office, European Research Center
Germany

Alois Knoll
Technische Universität München
Department of Informatics
Germany

***Abstract*—The "Providentia" system aims to enable highly automated vehicles, which are connected with the latest generation mobile networks, to obtain a far-reaching forward view on the relevant traffic situation by means of distributed high-performance sensors and real-time data fusion. The project aims to provide technologically pace-setting findings for next generation (5G) mobile networks, for the creation of highly reliable images of the reality in the infrastructure ("back end") with sensor fusion, for the supply of information to highly automated connected vehicles, for the virtualization of complex traffic scenarios, and – importantly – for the upcoming standardization of fifth generation mobile networks.**

Keywords— V2X, autonomous driving, connected cars, 5G

I. Introduction

An accurate perception of the environment is a key factor for safe manual or automated driving. An enhanced perception range would allow drivers or automated vehicles to estimate their necessary future actions more accurately. However, the perception range is often limited by multiple factors, such as other vehicles blocking the view ahead, adverse weather conditions and general sensor limitations.

Fig. 1. Artificial traffic scene illustrating perception range of individual automated vehicles on a highway in IPG CarMaker.

Latest advances in vehicle localization, available sensors and mobile network technologies for vehicle-2-everything (V2X) communication offer significant opportunities for improved cooperative perception, cooperative automated driving and the development of supporting platform systems, which provide necessary information to vehicles and drivers.

The goal of project "Providentia – proactive video-based use of telecommunications technologies in innovative motorway scenarios" is to significantly increase the perception range of vehicles and drivers in highway scenarios with a combination of data fusion and communication through the latest generation mobile networks. A comprehensive look at the road ahead is provided in a reliable, situation-adapted manner, while information overload is avoided. The system is supposed to function in optimal conditions, as well as under adverse weather conditions. Parts of this technical report, which aims to present the project, are based on the recent Providentia press release [1].

II. Related Work

The low-latency and high reliability of 5G in V2X enable new use-cases for cooperative automated driving [2]. Simultaneously, use-cases for automated driving have a strong influence on the requirements for 5G systems [3]. Large scale test bed implementations, such as the development of the far-reaching forward view in a digital test bed in the Providentia project, provide valuable data for the derivation of requirement for and the verification of 5G systems.

The combination of a large variety of different sensors available in mixed traffic scenarios with infrastructure and vehicle sensors is further simplified by the development of new concepts for smart adaptive data aggregation [4], which facilitate the flexible integration of new sensor sources into data fusion processes. A number of data fusion architectures are available [5][6], while a broader range of these become applicable due to the low latencies and high bandwidths of 5G communication.

Additionally, the development of new approaches for cooperative vehicle-infrastructure localization [7] makes it possible to derive accurate position estimates with limited information, based on the measurements of multiple connected vehicles.

A variety of highway monitoring systems have been designed or implemented in the recent years to fulfill limited tasks [8][9]. The value of V2X for a wide range of use-cases has been highlighted in [10]. The simultaneous improvements of 5G, automated driving, machine learning and sensors now create room for significant improvements in new systems.

III. Providentia

Within the project, developments and large scale tests are carried out in the A9 test bed. A research goal is to investigate the interplay of various information streams in highly automated vehicles, as well as the communication and back-end infrastructure. Using data-fusion processes, a detailed model of the environment is created in the infrastructure's computer system: – a "real time digital twin".

The digital twin is the base for the far-reaching forward view that is provided to the automated vehicles and the drivers. Sensors, such as cameras and radars along the road, combined with sensors in the vehicles and additional information derived from the optimized, connected mobile network, provide the data input for the data fusion.

The goal of the project is to address fundamental issues of highly automated and connected vehicles, such that it is possible to immediately test corresponding solutions in the harsh reality of a real, heavily travelled motorway, instead of an isolated test cell with very few vehicles.

The developed system is directly applicable to mixed traffic scenarios, including both vehicles with a variety of sensor sets and degrees of automation. Such mixed scenarios will be of major importance for years to come. Therefore, this scenario will have a strong impact on the upgrade of the 4.5G communications infrastructure towards 5G in the future.

IV. Conclusion

From the project, we expect technologically pace-setting findings for next generation (5G) mobile networks, but also for the creation of highly reliable images of the reality in the infrastructure ("back end") with sensor fusion, for the supply of information to highly automated connected vehicles, for the virtualization of complex traffic scenarios, and – importantly – for the upcoming standardization of fifth generation mobile networks.

The Providentia project will therefore demonstrate how road safety and effectivity can be improved for everyone by combining modern infrastructure sensors, the next communication designs and sophisticated data fusion approaches. The resulting system will provide critical momentum to the introduction of highly automated vehicles.

The work on the project has already begun with the first tests being carried out. Initial demonstrations can be expected in about one year. The final presentation is expected in autumn 2019. It is planned to publish the results in suitable formats.

Acknowledgment

This research has been funded by the Federal Ministry of Transport and Digital Infrastructure.

Federal Ministry
of Transport and
Digital Infrastructure

References

[1] "Neues Forschungsprojekt "Providentia" auf dem digitalen Testfeld Autobahn", http://www.presseportal.de/pm/108888/3572475, 20.02.2017, checked 03.03.2017.

[2] Cao, Hanwen, et al. "A 5G V2X testbed for cooperative automated driving." Vehicular Networking Conference (VNC), 2016 IEEE. IEEE, 2016.

[3] Ma, Hui Sheng, et al. A V2X Design for 5G Network Based on Requirements of Autonomous Driving. No. 2016-01-1887. SAE Technical Paper, 2016.

[4] „Smart Adaptive Data Aggregation – Poject SADA", http://www.projekt-sada.de/, checked 03.03.2017

[5] Aeberhard, Michael, and Nico Kaempchen. "High-level sensor data fusion architecture for vehicle surround environment perception." Proc. 8th Int. Workshop Intell. Transp. 2011.

[6] Michael Munz „Generisches Sensorfusionsframework zur gleichzeitigen Zustands- und Existenzschätzung für die Fahrzeugumfelderkennung", Dissertation Universität Ulm, 08.07.2011

[7] Zhang, Feihu, et al. "Cooperative vehicle-infrastructure localization based on the symmetric measurement equation filter." GeoInformatica 20.2 (2016): 159-178.

[8] Mukherjee, A., et al. "Large animal detection and continuous traffic monitoring on highways." SENSORS, 2013 IEEE. IEEE, 2013.

[9] Alexiadis, Vassili, et al. "The next generation simulation program." Institute of Transportation Engineers. ITE Journal 74.8 (2004): 22.

[10] Cacilo, Andrej, et al. "Hochautomatisiertes Fahren auf Autobahnen– industriepolitische Schlussfolgerungen, Dienstleistungsprojekt 15/14, Studie im Auftrag des Bundesministeriums für Wirtschaft und Energie (BMWi)." (2016).