Tobias Meuser, Daniel Bischoff, Ralf Steinmetz, Björn Richerzhagen. *Simulation Platform for Connected Heterogeneous Vehicles*. In Proc. of the 5th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS), http://www.vehits.org/, ISBN: 978-989-758-374-2

Simulation Platform for Connected Heterogeneous Vehicles

Tobias Meuser, Daniel Bischoff, Ralf Steinmetz, and Björn Richerzhagen
Multimedia Communications Lab (KOM), Technische Universität Darmstadt, Darmstadt, Germany
{*firstName.lastName*}*@KOM.tu-darmstadt.de*

Keywords: Vehicular Networks, Simulation, Rapid Prototyping

Abstract: The increasing number of connectivity features in current vehicles poses additional challenges for large-scale vehicular communication systems. Already deployed systems rely on the cellular network infrastructure, while the Wifi-based 802.11p standard will likely be implemented on a large scale in the next years. As real-world tests are costly, simulations are used to develop mechanisms for efficient short-range communication via 802.11p. However, efficient long-range communication between vehicles is pivotal for non-safety related information sharing. Current simulators often focus on short-range communication exchange, while approaches for efficient long-range communication are barely considered in automotive scenarios.

To enable rapid development of new approaches, we propose a scalable simulation environment for automotive applications. Our contributions are (i) the realistic modeling of heterogeneous vehicles including sensors and network interfaces, (ii) the automated generation of road properties like accidents and jams, and (iii) a configurable back-end infrastructure distributing events to the vehicles. All of the above contributions enable rapid prototyping and evaluation of automotive applications in various environments. We showcase two exemplary use cases to demonstrate the versatility of our simulation framework: an efficient road-based dissemination approach for long-range information exchange and a distributed information validation approach.

1 Introduction

In recent years, increasingly complex Advanced Driver Assistance Systems (ADASs) were developed, which require vast amounts of sensor information (Tigadi et al., 2016). Vehicles exchange information with each other to improve their perception range and quality. As practical tests are often unsuitable for the vehicular scenario, network simulators are used to model this information exchange. These simulators commonly model the information exchange via 802.11p, the cellular network, or a combination.

Local Inter-Vehicle Communication (IVC) via 802.11p is required for research performed on collaborative perception using Cooperative Awareness Messages (CAMs) (ETSI 102 637-2 V1.2.1, 2011) or Decentralized Environment Notification Messages (DENs) (ETSI 102 637-3 V1.1.1, 2010), and information exchange in Vehicular Ad-hoc Networks (VANETs) in general (Riebl et al., 2015; Günther et al., 2015). This type of communication is used for both safety-related applications like accident prevention and non-safety-related applications like offloading of the cellular connection.

The cellular connection is mainly used for the exchange of non-safety related information like traffic information and road properties. If a vehicle receives information, it can detour to prevent a jam or decelerate in-time to prevent emergency braking. The efficient distribution of these messages is a challenging topic in research, as the context of the vehicle and the message need to be considered in the dissemination. For many non-safety related applications, an exact model of the channel like in (Virdis et al., 2015) is often not required, as a message drop can often be compensated on the higher layers.

However, the heterogeneity of the vehicles in terms of their available network interfaces, computational resources, and sensors is an essential aspect for all vehicular applications. Today's network simulators often focus only on one dimension of heterogeneity, which limits the possible simulation scenarios. Based on this gap, we design our simulation framework for the vehicular scenario which simulates heterogeneous vehicles. Additionally, we enable rapid prototyping of vehicular applications, as we provide networking, processing and sensing components to limit the additional implementation effort when evaluating new approaches. Our simulation framework is based on the Simonstrator framework (Richerzhagen et al., 2015) and the vehicular traffic simulator Simulation of Urban Mobility (SUMO) (Lopez et al., 2018). SUMO provides realistic movement of the vehicles, while the Simonstrator focuses on rapid prototyping of networking applications of various domains.

Our contributions for the simulation of vehicular scenarios are (i) the setup of the road network including the realistic movement of the vehicles, (ii) the generation events on the roads which can be detected by the vehicles, (iii) the configuration of each vehicle in terms of available sensors and network interfaces and (iv) the connectivity services utilized by the vehicles. For the movement of the vehicles, we couple the Simonstrator and SUMO bidirectionally using the Traffic Control Interface (TraCI) (Wegener et al., 2008), i. e., the Simonstrator can influence the vehicle movement. Based on the vehicle information gathered from SUMO, we configure each vehicle individually. This configuration includes different storage and computation capabilities, sensors, and network interfaces. Additionally, we provide different connectivity solutions, including a central back-end and a pure-forwarding based broker.

All of the above contributions enable rapid prototyping and evaluation of automotive applications under varying environmental conditions. The variety of available communication paradigms and methods in the Simonstrator like the Publish/Subscribe (Pub/Sub) paradigm (Richerzhagen et al., 2015), and different ad-hoc communication schemes and mechanisms for cellular offloading (Richerzhagen et al., 2016) further support rapid prototyping of automotive applications. Especially the Pub/Sub pattern is pivotal for the automotive scenario, as inter-vehicle communication is often information-centered (Amadeo et al., 2016).

The rest of this work structures as follows: In Section 2, we provide an overview of the existing simulators for the vehicular scenario. Next, we describe our first contribution, the modeling of the environment based on the Simonstrator in Section 3. This modeling includes the road network, the vehicles, and the generation of road events. In Section 4, we describe the different connectivity configurations we provide in our automotive simulation framework in detail. Next, we show two exemplary use cases in the vehicular scenario in Section 5. Finally, we conclude our paper in Section 6.

2 Related Work

Several network simulators like NS-3 (Riley and Henderson, 2010) and OMNeT++ (Varga and Hornig, 2008) aim to simulate vehicular communication. To simulate inter-vehicle communication, these simulators model the vehicles' movement and network communication. For the vehicle movement, SUMO (Lopez et al., 2018) simulates the vehicle movement for all these simulators. For the network simulation, the above simulators differ significantly regarding available network models and vehicle modeling. NS-3 with the RACE extension (Jomrich et al., 2017) can precisely model the cellular channel including X2-handover but focuses on the pure networking aspects of the vehicular scenario. Thus, the development of new applications is complex.

In that regard, OMNeT++ is much more versatile through the high number of available extensions. The extension Veins (Sommer et al., 2011) couples OMNeT++ with SUMO bidirectionally, such that OMNeT++ applications can influence the movement of the vehicles. While Veins initially provided only adhoc communication between vehicles, it was later extended to support the exchange of messages via Long Term Evolution (LTE) (Hagenauer et al., 2014). To follow current trends of the automotive industry, Artery (Riebl et al., 2015) added the ITS-G5 stack to Veins, which enable the test of novel VANET applications. In this approach, the vehicle's onboard sensors are modeled in an extendable manner. However, Artery did not consider the heterogeneity of sensors regarding sensor quality. Artery was further extended by local perception sensors and the LTE communication stack (Günther et al., 2017), which increased the possible vehicular applications further.

While all of the above approaches are suitable for their intended use cases, they lack at least one of the following properties which are essential for long-range communication in vehicular networks: the heterogeneity of the vehicles in case of the NS-3 extension or the lack of different cellular communication paradigms in case of OMNeT++. This lack increases the implementation time of vehicular applications and slows down the development of new features.

Thus, we base our automotive evaluation setup for non-safety-related applications on the event-based Simonstrator (Richerzhagen et al., 2015), which supports rapid prototyping in different scenarios. While the Simonstrator only supports a limited amount of channel models, it provides the possibility to use different communication paradigms and node sensors. Consequently, we focus more on applications and the improvement of message dissemination and filtering. To this end, we model the vehicle including vehicle sensors, network interfaces, computational resources, and different back-end communication strategies like pull-based and push-based communication.

Different back-end communication strategies are

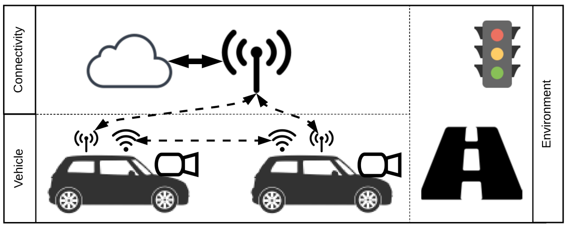


Figure 1: Overview of the required components for the modeling of vehicles.

especially valuable for the evaluation of vehicular applications. Currently, these applications rely mostly on a central server entity, which is continuously queried for new information (Turcanu et al., 2016; Rayeni et al., 2018). However, the monetary costs for server maintenance might hinder their practical applicability. These approaches might be enhanced further by utilizing highly-scalable low-resource communication paradigms like Pub/Sub (Gascon-Samson et al., 2015; Majumder et al., 2009), but due to the restrictions of the existing simulators, these approaches cannot be evaluated. Our automotive evaluation setup resolves this issue and enables the variation of different back-end communication strategies.

3 Environmental Modeling

Figure 1 displays the required components for a network simulator in the vehicular scenario. This scenario consists of vehicles, connectivity features, and the environment. Each vehicle is equipped with several sensors and networking capabilities. A central back-end may coordinate the message dissemination and offer additional back-end services. As the sensors measure the environment, we need to model the environment first. Section 3.1 describes the modeling of the environment including the efficient accessibility of the road network in the network simulator and the extensible generation of measurable road properties. Subsequent, Section 3.2 describes the modeling of the vehicle including the vehicle movement, available sensors, network interfaces, and storage capabilities. Using the components from these two sections, Section 4 describes the connectivity, which includes different communication paradigms, ad-hoc and cellular dissemination approaches, and offloading schemes.

3.1 Environment

For the vehicular scenario, information is commonly geocasted to a particular area. Geocasting often relies solely on the location of each vehicle. Thus, no information about the road network is required. In recent years, pure location-based geocasting has proven to be inefficient, as vehicle follow predictable routes. If information about the road network is used in the dissemination process, the cellular traffic decreases drastically (Meuser et al., 2018b). To this end, the road network needs to be accessible to develop approaches that improve communication quality. We retrieve the road network from SUMO using TraCI.

The road network class is accessible from every class in the Simonstrator and manages all roads in the scenario. A road consists of a unique id, the maximum allowed speed, its length, the corresponding lanes, incoming and outgoing roads, and active properties. The incoming and outgoing roads are required to enable rapid path search. Additionally, the active properties describe the current state of the road.

3.1.1 Road Property Modeling

We model a road property if it supports the driver in his driving decisions and, thus, is shared between vehicles. It provides the creation date of the property, the value of the property (if necessary), and the default value of the property, e. g., *no jam* in case of a jam property. Currently available properties are fog, bump, hazard, jam, rain, and traffic signs. However, new properties can be added easily due to the extensible design of our environment properties.

Some properties like jams influence the behavior of the vehicles directly. Thus, these properties change the representation of the road in SUMO. In that case, the road is manipulated using TraCI to set the corresponding properties like lowering the allowed road speed. If only start and destination are provided, this leads to issues in SUMO if shortest-path routing is utilized. Due to the changes of the road speed, the vehicle's route might be adjusted. However, this route change is based on global knowledge, which would not be available in real-world scenarios. Thus, only scenarios like the TAPAS Cologne scenario shall be used currently, as the routes are predefined there.

In our vehicular scenario, the location of road events has a significant influence on the system performance. As the properties are generated artificially, the generation is essential for the simulation results.

3.1.2 Road Property Generation

At the start of the simulation, no road has an active property, i. e., no roads are jammed or otherwise affected. To this end, the vehicles do not need to communicate. After a configurable time, the generation of road properties starts. Once a property is created, the

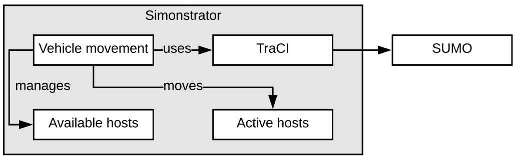


Figure 2: Connection of the Simonstrator and SUMO.

location, *duration*, *type*, and the *value* of this property need to be specified. As these parameters influence the simulation results, our property generator is extensible by multiple plugins. Each plugin is responsible for the generation of a particular road property. These plugins choose the *location* of the property randomly. This randomness can be configured to either induce a specified load to the network or reproduce real-world behavior based on statistical observations.

Regarding the *lifetime* of a property, we offer different possible lifetime distributions to simulate different environmental conditions. We designed three lifetime distributions for our scenario, (i) an exponential distribution, (ii) a Gaussian distribution, and (iii) a custom distribution to reflect real-world measurements. Each distribution can be configured according to the type of the property that is simulated. Once the event lifetime is reached, the property value changes.

Depending on the chosen generator and the event *type*, the behavior of the property *value* might be either continuous or discrete. An example of a continuous variable is the driving speed in a jam, while a hazard is a discrete variable with the states *hazard* and *no hazard*. For discrete variables, we use a Markov chain to determine the transitions between the states of the variable. For continuous variables, we also divide them into states of similar size, with a transition probability between the states. Between the state changes, the variable value is linearly interpolated between the two states. To simulate different types of events, we have one property generator plugin per event type. Thus, it is possible to plugin custom property generators. That way, new road properties can easily be added to our simulation.

3.2 Vehicle

Vehicles can only sense road properties using their equipped sensors. As the range of these sensors is limited, the movement of the vehicles impacts the measurable road properties. Thus, realistic vehicle movement is required for reliable simulation results.

3.2.1 Vehicle Movement

Map-based movement is a common characteristic of vehicle movement. For map-based movement, the Simonstrator targets only human mobility (Richerzhagen et al., 2017), which does not satisfy the requirements of realistic vehicle movement. Thus, we connect the Simonstrator to the traffic simulator SUMO to achieve realistic vehicle movement. Figure 2 displays the technical implementation of the connection. The *vehicle movement* class utilizes TraCI to access the information available at SUMO. Based on this information, the *vehicle movement* moves the vehicles in the scenario. An issue regarding the connection of the Simonstrator and SUMO is the fluctuating number of active vehicles in SUMO. Due to simulator restraints, we cannot generate new hosts at runtime. To resolve this issue, we initialize a number of offline hosts which are initialized at the start of the simulation. This number depends on the scenario chosen in SUMO. Once a new vehicle spawns in SUMO, the *vehicle movement* selects one host from the set of available hosts and turns its network interfaces online. Similarly, the network interfaces are turned offline after the vehicle went inactive in SUMO. The *vehicle movement* binds the selected host to the respective vehicle in SUMO. In our simulation framework, every application can read information from SUMO and manipulate the vehicle movement in SUMO. Examples for information are the future route of the vehicle and the current vehicle speed. To dynamically exchange the scenario through different simulation runs, we mirrored the essential parts of the configuration of SUMO in the Simonstrator. As simulating the whole network might not be necessary for huge scenarios, we provide the possibility to reduce the size of the simulated area independently of SUMO. Only if a vehicle enters the described area of the scenario, it is modeled in the Simonstrator, and its network interfaces and sensors are activated.

3.2.2 Sensors

For sensing the environment, we model multiple sensors for the vehicular use case. Compared to previous sensor modeling approaches like in (Günther et al., 2017), we explicitly model sensor inaccuracy in our system. This inaccuracy is important for the vehicular use case, as perfect information will not be available. Thus, every sensor has an accuracy value, which states the probability that the sensor measures the correct value. We added a configurable random deviation to the sensor quality to reflect the sensor quality difference of vehicles in real-world conditions.

In this work, we can simulate both continuous and discrete events. As each continuous variable can be approximated with a discrete variable, we perform most of our evaluations on discrete variables to reduce complexity. To this end, we divide the values of the variable into *n* possible states. For continuous variables, *n* should be chosen such that a required accuracy level is reached.

There are different sensor results available: (i) a single value that can be measured for both continuous and discrete variables, and (ii) a probability vector that is only available for discrete variables. This probability vector states the probability that the variable is in a particular state. The advantage of the probability vector is the provided meta-information about the inaccuracy of the sensor. This meta-information is important in the aggregation process, as the impact of low-quality sensors is reduced.

When a sensor measures the environment, it needs to determine the measured state. This state depends on the final accuracy of the sensor, i. e., the accuracy after the random deviation was added. To reflect different types of environmental variables, we modeled the sensing process using different distributions. However, for most use cases, the Gaussian distribution is considered to be most appropriate. Compared to the work of (Meuser et al., 2018c), we revised the design of the sensors, which simplifies the configuration of sensors for different use-cases. Additionally, new sensors can easily be added through our modular sensor design. Similarly, new road properties can be added easily through our extensible design. To provide a basic set of properties, we implemented sensors for fog, jam, accident, bump, and traffic signs.

Every simulated automotive application can query the current observations from the available sensors. The observations of all plugins are bundled into an *Environment* object. Depending on the application requirements, only some sensor plugins may be required. Thus, the deployed plugins can be configured either on startup or during the request.

At the simulation start, each vehicle is provided with a configurable set of sensors with configurable quality. This configurability is pivotal for vehicular networks, as the heterogeneity of the sensor network is a common topic for vehicular applications. While the heterogeneity of vehicular sensors is one important aspect, vehicles are additionally heterogeneous regarding their available computation capabilities, storage, and network resources.

3.2.3 Network Resources and Computational Resources

Vehicles have the possibility to exchange information via two network types, decentralized and centralized communication. In our simulation, we assume that the decentralized communication is performed Wifi-based and the centralized communication is performed based on the mobile network. While Wifi-based communication is commonly used for CAMs, it can also offload the cellular connection or carry messages over large distances. However, mobile communication is often more suitable for information exchange over large distances. As both communication types are viable in the vehicular scenario, we provide the possibility to use both decentralized and centralized communication in the vehicles.

We wrapped the road properties from above into the transmittable *RoadInformation* class which contains a RoadProperty and additional attributes like expected event lifetime. The RoadProperty contains the respective road id, the measurement location, the date of the measurement, and the value of the property. With the help of this class, communication is independent of the concrete road property implementation. Dependent on the road property, often one of the two communication technologies is preferable. For jam information, LTE is preferable, as the jam information required early to potentially detour. Contrary, information about a bump can be disseminated via 802.11p, as the driver only needs to slow down. However, not all vehicles can communicate via both of these technologies. Thus, a single technology is often not sufficient for vehicular communication. To design applications solving this issue, we configure the network interfaces for each vehicle individually. Depending on the available communication technologies, the Simonstrator provides different communication patterns. These include the Pub/Sub communication paradigm (Richerzhagen et al., 2015), ad hoc dissemination mechanisms and gateway selection strategies (Richerzhagen et al., 2016). With the availability of these patterns, new applications can easily be evaluated without consideration of the underlying communication paradigm. The possibilities for communication are discussed in Section 4.

Every time vehicles exchange measurements, they either use the received measurements immediately for decision-making or store them locally. While the immediate use of measurements requires only a few resources, the storing of information requires storage capabilities of the vehicles. While the limitation of storage seems to be negligible in this scenario, the huge number of roads with possibly important information might lead to storage issues. Thus, we modeled the cache of the vehicle with different sizes and invalidation strategies. As measurements might be inconsistent, the vehicles need to validate the received information to use it in their decision-making. We encapsulate the validation of information such that it is independent of the road property and uses only commonly available meta-information like location, detection date, and Time to Live (TTL). Based on this information validation, further investigation of optimal vehicular decision-making is possible.

4 Communication Components

Connectivity is an essential feature for future vehicular applications. While today's network simulators focus on the exact modeling of the data transmission, the overlay network is mostly ignored. An overlay network is on top of the underlay network (LTE, 802.11p) and provides different addressing schemes, e. g., content-based addressing. Especially, content-based information dissemination matches the requirements of vehicular applications well (Amadeo et al., 2016). To this end, our goal is to provide different types of overlay networks to kick-start the development and evaluation of vehicular applications.

4.1 Addressing Schemes

For the inter-vehicle communication, overlay networks with different addressing schemes are available. All these schemes consider the location as a pivotal property for the dissemination process. We implement existing schemes for addressing based on the exact vehicle location, geohash[1], street segments, and an information-dependent relevance score.

In the addressing based on the exact vehicle location, each vehicle receives all events in a particular area around its current location. While this addressing scheme is commonly used for Mobile Ad-hoc Networks (MANETs), it is unsuitable for vehicular networks due to the high node mobility. This forces frequent location updates and, thus, induces unnecessary load to the network. Based on this consideration, we implemented the geohash concept in the Simonstrator. The geohash differs from the addressing scheme based on exact location, as each vehicle only subscribes to the grid cell it is currently in. The size of the subscription area can be adjusted with increasing vehicle speed, which further reduces the number of location updates. However, even this addressing scheme induces a considerable overhead: Information is often not relevant in the whole grid cell, but only for a small set of roads in that cell. To alleviate for this issue, we implemented a road-based addressing scheme. In this addressing scheme, vehicles subscribe to street segments and receive the information of these segments. The subscriptions to street segments reduce the overhead compared to the addressing based on exact location. However, the overhead is higher compared to the geohash approach, which is justified by the more efficient distribution of payload messages. For road-based addressing, each application must specify the set of roads, for which the information is relevant. In addition to this, we provide a relevance-aware addressing scheme which disseminates measurements depending on the context of both the measurement and the vehicle. The relevance-aware addressing scheme simplifies the configuration of the dissemination strategy drastically, as the application only needs to specify a threshold for the relevance between 0 and 1. The availability of these addressing schemes decreases the development time of new vehicular applications and the evaluation time of new addressing schemes drastically. Moreover, these strategies are applicable to every communication strategy, which are presented in the following.

4.2 Communication Strategies

There are two communication strategies, pull-based communication, and push-based communication. Both communication strategies are suitable for local and cellular communication and have their specific challenges and advantages. Pull-based communication relies on periodic pulling of information, which induces additional network load for the requests and, thus, reduces the available bandwidth for payload messages. Push-based communication omits these requests, but might share information that is not required by the receiver.

The normally used strategy differs for the underlying communication technology. Local communication via 802.11p is commonly push-based, as the currently available messages like CAM and Decentralized Environment Notification (DEN) are sent periodically or event-based. Thus, pull-based communication would increase bandwidth usage without benefiting the network performance. Cellular communication is often performed pull-based, as current vehicular applications require only a few data, which leads to few pull requests. A central server manages all available information and distributes them to the vehicles. Future vehicular applications will require an increasing amount of information. Thus, the load on this central server and the cellular network will increase. Push-based communication generally decreases the load on the server, as less server storage and computational power is required. This load decrease is due to the local processing of information in the vehicles. Additionally, no requests like in pull-based communi-

---

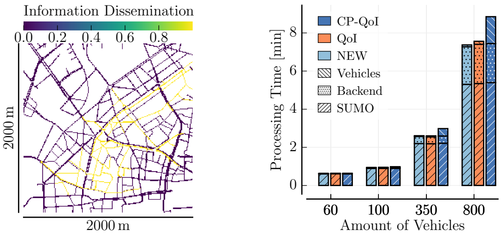
[1] http://geohash.org/

cation are required, which might decrease the overall consumed bandwidth. To model both current and future communication scenarios, we implemented both communication strategies, including strategies combining push-based and pull-based communication. To further reduce the load on the cellular network and compensate for dead spots, we implement strategies for offloading the cellular connection based on Wifi. The required server configuration is described in the following section.

4.3 Backend Configuration

The required amount of server resources depends on the used communication strategy. While pull-based communication generally requires computational power and storage, push-based communication only requires information about the vehicle's positions. Thus, the required server resources for push-based communication are generally lower compared to pull-based communication, as vehicles manage and aggregate provided information.

For pull-based communication, the server often aggregates information centrally and distributes the aggregates to the vehicles. However, relying only on a central server limits the scalability and contradicts the edge computing trend for mobile networks. If the server resources are not sufficient or the server costs should be minimal, push-based communication is generally preferable. A push-based server without storage provides information of lower quality to the vehicles, which the vehicles aggregate locally using their computational resources. Additionally, a hybrid strategy is available, in which information is distributed pull-based or push-based dependent on different information properties. As the performance of these approaches depend on the available server resources, we provide the possibility to configure the server regarding available storage.



(a) Exemplary information dissemination range for a threshold of 1% in (Meuser et al., 2018b).

(b) Required processing for the information validation approaches in (Meuser et al., 2018a).

Figure 3: Additional insights to our approaches of our previous work.

5 Exemplary Use Cases

In this section, we provide an overview of our previous work, which used our simulation platform (Meuser et al., 2018b; Meuser et al., 2018a). Due to space constraints, the reader is referred to the respective papers for an in-depth description of the evaluation results. The difference in application scenarios shows the versatility of our simulation framework which is freely available[2]. To support the evaluation of developed approaches, we designed different metrics assessing the behavior of developed approaches from different perspectives. Examples of available metrics are the quality of information at the vehicles, the produced network traffic, and the computational effort of the approaches.

In (Meuser et al., 2018b), we evaluated our approach with a focus on information validation for vehicular applications. Thus, we used vehicles with a default communication interface to speed the development process of their information validation approach and focused on information quality metrics. Figure 3(b) shows the computational overhead of the different approaches, which can be useful to evaluate the practical applicability of the developed information validation approaches.

In (Meuser et al., 2018a), we focused on a dissemination strategy, which aims to distribute information efficiently while preserving communication quality. Thus, we used the available information validation modules to concentrate on the networking aspect of the investigated problem and used network traffic metrics. Figure 3(a) displays the communication range of one specific message. This visualization can be very useful to analyze the impact of different environmental conditions on the dissemination and support the development of new dissemination approaches.

6 Conclusion

In this work, we present a platform for the rapid development and evaluation of automotive networking applications. Compared to existing simulators, our platform focuses on the heterogeneity of the vehicles under varying environmental conditions. For the vehicles, this includes different sensors, storage capabilities, and networking components. For the environment, we simulate different types of measurable

---

[2] dev.kom.e-technik.tu-darmstadt.de/simonstrator/

events. Additionally, we provide different server configurations to evaluate varying networking conditions.

We demonstrate the versatility of our platform by evaluating different types of automotive applications. While the focus of both applications differs, our platform can be used to evaluate both applications. Additionally, our platform is used by many researchers and continuously extended. Currently, we are extending our platform with strategies for offloading the cellular connection and efficient event dissemination.

ACKNOWLEDGEMENTS

This work has been funded by the DFG within the CRC 1053 - MAKI (B1, C2).

REFERENCES

Amadeo, M., Campolo, C., and Molinaro, A. (2016). Information-centric networking for connected vehicles: a survey and future perspectives. *IEEE Commun. Mag.*, 54(2):98–104.

ETSI 102 637-2 V1.2.1 (2011). Part 2: Specification of Cooperative Awareness Basic Service. Standard, European Telecommunications Standards Institute, Sophia Antipolis Cedex, FR.

ETSI 102 637-3 V1.1.1 (2010). Part 3: Specifications of Decentralized Environmental Notification Basic Service. Standard, European Telecommunications Standards Institute, Sophia Antipolis Cedex, FR.

Gascon-Samson, J., Garcia, F.-P., Kemme, B., and Kienzle, J. (2015). Dynamoth: A Scalable Pub/Sub Middleware for Latency-Constrained Applications in the Cloud. In *Proc. of IEEE ICDCS*. IEEE.

Günther, H. J., Timpner, J., Wegner, M., Riebl, R., and Wolf, L. (2017). Extending a holistic microscopic IVC simulation environment with local perception sensors and LTE capabilities. *Vehicular Communications*, 9.

Günther, H.-J., Trauer, O., and Wolf, L. (2015). The potential of collective perception in vehicular ad-hoc networks. In *Proc. of IEEE ITST*. IEEE.

Hagenauer, F., Dressler, F., and Sommer, C. (2014). Poster: A simulator for heterogeneous vehicular networks. In *Proc. of IEEE VNC*. IEEE.

Jomrich, F., Wankhede, T., Ruckelt, T., Burgstahler, D., Bohnstedt, D., and Steinmetz, R. (2017). Demo: rapid cellular network simulation framework for automotive scenarios (RACE framework). In *Proc. of IEEE NetSys*. IEEE.

Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., and Wießner, E. (2018). Microscopic traffic simulation using sumo. In *Proc. of IEEE ITSC*. IEEE.

Majumder, A., Shrivastava, N., Rastogi, R., and Srinivasan, A. (2009). Scalable Content-Based Routing in Pub/-Sub Systems. In *Proc. of IEEE INFOCOM*. IEEE.

Meuser, T., Lieser, P., Nguyen, T. A. B., Böhnstedt, D., and Steinmetz, R. (2018a). Lifetime-Aware Information Aggregation under Uncertainty for Advanced Driver Assistance Systems. *Ad Hoc Networks*.

Meuser, T., Richerzhagen, B., Stavrakakis, I., Nguyen, T., and Steinmetz, R. (2018b). Relevance-Aware Information Dissemination in Vehicular Networks. In *Proc. of IEEE WoWMoM 2018*. IEEE.

Meuser, T., Wende, M., Lieser, P., Richerzhagen, B., and Steinmetz, R. (2018c). Adaptive Decision Making based on Temporal Information Dynamics. In *Proc. of VEHITS*. SciTePress.

Rayeni, M. S., Hafid, A., and Sahu, P. K. (2018). Quality of service aware multicasting in heterogeneous vehicular networks. *Vehicular Communications*, 13.

Richerzhagen, B., Stingl, D., Rueckert, J., and Steinmetz, R. (2015). Simonstrator: Simulation and Prototyping Platform for Distributed Mobile Applications. In *Proc. of EAI SIMUTOOLS*. ACM.

Richerzhagen, N., Richerzhagen, B., Stingl, D., and Steinmetz, R. (2017). The human factor: A simulation environment for networked mobile social applications. In *Proc. of IEEE NetSys*. IEEE.

Richerzhagen, N., Richerzhagen, B., Walter, M., Stingl, D., and Steinmetz, R. (2016). Buddies, not enemies: Fairness and performance in cellular offloading. In *Proc. of IEEE WoWMoM*. IEEE.

Riebl, R., Günther, H.-J., Facchi, C., and Wolf, L. (2015). Artery: Extending Veins for VANET applications. In *Proc. of IEEE MT-ITS*. IEEE.

Riley, G. F. and Henderson, T. R. (2010). The ns-3 Network Simulator. In *Modeling and Tools for Network Simulation*. Springer Berlin Heidelberg, Berlin, Heidelberg.

Sommer, C., German, R., and Dressler, F. (2011). Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis. *IEEE TMC*, 10(1).

Tigadi, A., Gujanatti, R., and Gonchi, A. (2016). Advanced Driver Assistance Systems. *iJERGS*, 4(3).

Turcanu, I., Sommer, C., Baiocchi, A., and Dressler, F. (2016). Pick the right guy: CQI-based LTE forwarder selection in VANETs. In *Proc. of IEEE VNC*. IEEE.

Varga, A. and Hornig, R. (2008). An overview of the OMNeT++ simulation environment. *Proc. of SIMUTOOLS*.

Virdis, A., Stea, G., and Nardini, G. (2015). Simulating LTE/LTE-Advanced Networks with SimuLTE. In *Simulation and Modeling Methodologies, Technologies and Applications*. Springer.

Wegener, A., Piorkowski, M., Raya, M., Hellbrck, H., Fischer, S., and Hubaux, J.-P. (2008). Traci: An interface for coupling road traffic and network simulators. *Proc. of CNS*.