A V2X Message Evaluation Methodology and Cross-Domain Modelling of Safety Applications in V2X-enabled E/E-Architectures

Harald Bucher, Marius-Florin Buciuman, Alexander Klimm, Oliver Sander, Juergen Becker
Karlsruhe Institute of Technology (KIT)
Institute for Information Processing Technology (ITIV)
Engesserstrasse 5
76131 Karlsruhe, Germany
{bucher, alexander.klimm, oliver.sander, becker}@kit.edu

ABSTRACT

The introduction of Vehicular Ad-Hoc Networks (VANETs) enables great potential for improving road traffic flow and especially active safety applications such as cooperative adaptive cruise control (CACC). Such applications not only rely on continuous broadcast of vehicle state information (beacons) of all vehicles, but also have strict real-time requirements. Regarding automotive E/E architectures this continuous broadcasting adds heavy internal E/E data traffic that needs to be processed in real-time by Electronic Control Units (ECUs). In this work we address this issue by proposing a novel cluster-based message evaluation methodology to significantly reduce internal E/E network traffic by discarding irrelevant messages. The approach is only depending on information received over beacons. It combines a vehicle clustering strategy as well as network and vehicle state monitoring capabilities in order to correctly evaluate messages under real-time constraints. The proposed methodology is modeled inside an abstract ECU. It is evaluated by simulating a model-based CACC application under different traffic scenarios. It is shown that a significant reduction of messages is achievable, while still guaranteeing accident-free behavior of CACC.

Categories and Subject Descriptors

I.6.2 [**Simulation and Modeling**]: Simulation Languages; I.6.5 [**Simulation and Modeling**]: Model Development; I.6.3 [**Simulation and Modeling**]: Applications

General Terms

Design, Languages

Keywords

V2X Message Evaluation, WAVE, E/E-Architectures, Co-Simulation, Heterogeneous Modelling

1. INTRODUCTION

Today, the electric/electronic (E/E) architecture of modern cars is a distributed network of embedded systems, consisting of several bus systems, dozens of electronic control units (ECUs) and hundreds of sensors and actuators. Due to the continuous rising number of functions, current E/E architectures are evolving towards new design principles like standardization, encapsulation and centralization [1]. This situation is aggravated by the integration of new technologies like Vehicle-to-X communication (V2XC). V2XC forms the basis for improved safety, efficient traffic management and infotainment. Communication between vehicles in such vehicular networks is referred to as *Dedicated Short Range Communication* (DSRC). DSRC is based on IEEE standards 802.11p [2] and 1609 *Wireless Access in Vehicular Environments* (WAVE) [3] protocol stack. For active safety applications like *Cooperative Adpative Cruise Control* (CACC), it is assumed that every vehicle is equipped with a V2X ECU, periodically broadcasting its current state containing position, speed, acceleration, and direction to its surrounding neighbors as *beacons*. A beacon's format is standardized by [4] and also called *Basic Safety Message* (BSM). Especially at high traffic densities with hundreds of vehicles in range, this results in an enormous message appearance per time unit. It increases not only wireless channel load but also signature verification efforts and communication latencies in internal E/E networks [5]. The latter is crucial for safety-critical functions like CACC, which rely on information updated periodically [6].

One challenge often addressed within this context is controlling the load of the wireless channel to reduce packet losses and increase reception rate especially of safety related packets. Nevertheless, the received packets still need to be processed and routed through the internal E/E architecture. The latter is typically a priori designed as a closed system not capable to process and route the additional V2X messages. Another problem often addressed is fair data dissemination in the whole network which is, however, not suitable for safety applications, since it could harm real-time requirements of individual applications. Overall, most recent works look at a vehicle as a concentrated point, neglecting internal E/E architecture processing and communication efforts. Therefore, in [7] we developed an extensible tool chain that targets exploration, validation and verification of modern V2X-enabled

automotive E/E architectures. Especially it supports heterogeneous model composition and distributed co-simulation of E/E architecture models together with its environment and active safety-critical applications, e.g. CACC. Fig. 1 shows a typical (simplified) E/E architecture that can be found in current vehicles.
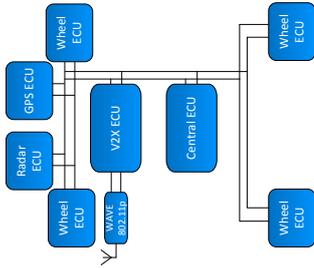

Figure 1: V2X-enabled E/E Architecture

Concurrently considering E/E characteristics with prioritized data evaluation and selection for safety-critical applications in order to reduce internal E/E processing and communication efforts are regarded insufficiently so far. Therefore, we contribute in this paper to this issue as follows: I) We propose a novel *cluster-based and WAVE compliant V2X message evaluation strategy for V2X ECUs* especially II) *taking into account E/E architecture artifacts* in order to reduce internal E/E network traffic and III) *model-based cross-domain evaluation of a CACC application in combination with E/E architecture components*.

Considering I) we introduce a classification of surrounded vehicles in *Farfield Network* (FFN) and *Nearfield Network* (NFN) to evaluate incoming beacons in a pipelined manner. Thereby, the approach is fully decentralized by relying only on data contained in beacons. Since we're focusing on safety-critical applications, the message evaluation is based on *WAVE Short Messages* (WSMs), because of their minimized protocol overhead optimized for safety purposes (see Sec. 2).

Furthermore with II) we consider E/E architecture aspects by providing metrics for monitoring network load in order to properly adapt message acceptance policies.

Considering III) we show that a significant reduction of messages that need to be processed during run time is achievable. The evaluation is based on the co-simulation tool chain developed in [7]. We use a heterogeneous model-based CACC application embedded in an abstract model of an E/E architecture (cp. Fig. 1) taking into account internal E/E processing and communication latencies.

The remainder of this paper is organized as follows: In Sec. 2 a brief review of DSRC is given. Sec. 3 discusses a selection of related works. The proposed message evaluation strategy is detailed in Sec. 4. Afterward, the approach is evaluated with selected case studies in Sec. 5. Sec. 6 gives a short discussion. Finally, Sec. 7 concludes and presents some future work.

2. FUNDAMENTALS

DSRC is based on the IEEE 802.11p standard [2]. It is an extension to the traditional IEEE 802.11 WiFi standard specifying the PHY and MAC layer to meet the requirements in vehicular environments. IEEE 1609 WAVE [3] family standards incorporate 802.11p, facilitating support for multi-channel access on the MAC layer (IEEE 1609.4) as well as higher layer services like security and networking. One control channel (CCH) is defined for system control and safety messages, whereas non-safety and high bandwidth traffic (e.g. UDP traffic) can be transmitted over four to six service channels (SCH) in Europe and in the U.S. respectively. Furthermore, WAVE provides a minimized protocol overhead to reduce latency and simplify joining and leaving basic services. Therewith, a new message type and protocol was introduced in IEEE 1609.3, the WSM and *WAVE Short Message Protocol* (WSMP) respectively. WSMP also supports a frame based adjustable power and data rate/channel as well as priority and service identifier (PSID) assignment allowing cross-layer optimizations. To increase interoperability between DSRC applications, application layer standards like SAE J2735 [4] are also incorporated in the overall protocol stack. It defines a message set dictionary for DSRC applications to provide a common language.

3. RELATED WORK

Since the introduction of VANETs together with its standards and cooperative applications many approaches were proposed addressing the control of the wireless channel utilization and improving traffic management and efficiency. Considering wireless channel control in [8] the authors use a combined power-rate adaption of beacon transmission based on estimated tracking error of surrounded vehicles and wireless channel occupancy in order to improve dissemination of safety related data. The authors in [9] build neural networks to efficiently adapt the beacon rate, but they rely on global information provided by Road Side Units. In [10, 11] they also make use of an adaptive beaconing protocol based on channel quality and message priority. They furthermore provide a protocol based on utility functions to optimize overall data utilization benefits of arbitrary applications. The latter, however, is not suitable for cooperative safety applications, since it could harm individual applications' real-time constraints [6].

Clustering mechanisms of vehicles are popular approaches for efficient data dissemination. [12] and [13] provide complex clustering algorithms but they mainly focus on stabilization of clusters itself in order to generalize traffic information and aim at trust modelling respectively. This approach is not suitable in our context of safety related message evaluation which demands for real-time constraints and lightweight computation complexity.

The emphasis of all previous works lie rather on analysis on a more coarse grained level like traffic management, transmission efficiency and overall data utilization. Because of that, in these works vehicles can be considered as a focused processing point. For validation of E/E architecture components, such a point-of-view is insufficient. There is rather the need for a comprehensive description of a complete V2X processing chain starting from the sensor of the source vehicle and ending up at the actor of the destination vehicle. Especially, they are not considering the receiving path of beacons. Also the related countermeasures for controlling internal E/E processing and communication efforts are not addressed. To the best of our knowledge, this work is the first one considering these issues and adheres to real-time

constraints that are mandatory for safety applications.

4. V2X MESSAGE EVALUATION DESIGN

In order to allow access to internal E/E artifacts and especially to the V2X radio interface for evaluating incoming messages it seems to be likely to realize the methodology as an intelligent gateway in a dedicated V2X ECU responsible for V2X related data processing. The proposed methodology is modeled within an abstract ECU model as illustrated in Fig. 2 and is mainly made up of five components: 1) *DSRC/WAVE Interface* for decoding and encoding WSMs including BSM 2) *E/E architecture interface* providing data to/of other ECUs (e.g. sensor and GPS data of local/remote vehicle) 3) *WSM Scheduler* for efficient selection and transmission of local application data or forwarding of safety critical incoming WSMs 4) *Hierarchical Message Evaluation Stages* as core component for data fusion and message evaluation/filtering 5) *Network Monitoring* capabilities allowing evaluation of both WSM channels and internal E/E network utilization.
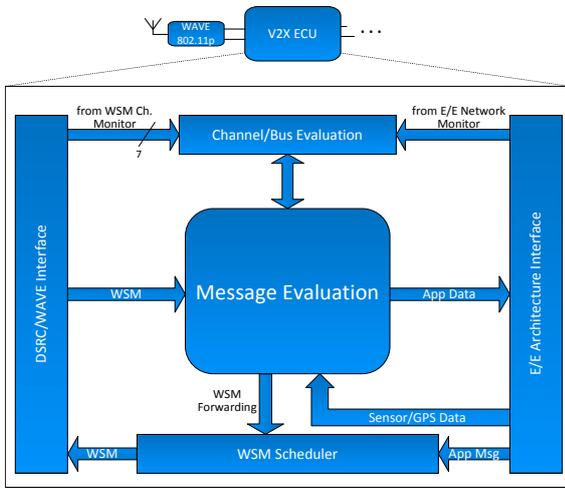

Figure 2: V2X Message Evaluation System Design

Since we're focusing on evaluating and filtering incoming WSMs, *WSM Scheduler* is not regarded in the following. Briefly, the *WSM Scheduler* implements a simple round robin scheduling of incoming application data from other ECUs and crucial WSMs from the evaluation stages. However, it can be seen as generic building block within the overall system design and could be refined by relevant and more complicated beaconing and dissemination approaches like [8] or [10].

In order to evaluate incoming messages properly, a comprehensive view of the environment and traffic situation is necessary. Based on position, speed etc. received over BSMs from surrounding vehicles, it is possible to construct a virtual network which is updated regularly. Two basic approaches exist to build up such virtual networks: A_I, every vehicle within radio range is used for considering it in evaluation process, and A_{II}, a fully cluster-based approach where vehicles with similar driving states (for instance position, speed, direction) are clustered. Thereby only messages from *cluster masters* are propagated to other clusters [12, 13], whereas data from *cluster slaves* are not. A_I provides highest accuracy in terms of the virtual environment, since every vehicle

in range is considered. At the same time it demands for higher processing power and memory capacity on an ECU. A_{II} is more flexible and scalable, it reduces message appearance but is not suitable for safety-critical purposes, since it is not ensured that the actual master holds the most relevant data for every application. Therefore we propose an ad-hoc cluster-based message evaluation scheme.

4.1 Ad-Hoc Cluster-based Message Evaluation

We propose a hybrid cluster-based message evaluation methodology by using a classification into a *Farfield Network* (FFN) and *Nearfield Network* (NFN) database. This adheres to combine advantages of both approaches A_I and A_{II}. NFN considers all vehicles in close proximity (see Sec. 4) by a parametrized position range, mainly dependent on current vehicle state. All vehicles within that range which fulfill the current acceptance policy becomes a node, i.e. a master in NFN. All messages are considered for evaluation, since they potentially contain information for safety-critical situations and applications, e.g. CACC. In contrast, *FFN* uses a cluster-based approach, where vehicles beyond the NFN area are clustered and are represented by a cluster master which in turn has its own NFN. Since information from FFN is potentially used for long-term optimizations or warnings (e.g. premature reaction to an accident) only messages sent by FFN masters which fulfill the current acceptance policy are considered for evaluation. Messages from slaves of the individual clusters can be discarded inherently. Masters are determined by simple *First Come First Serve* principle. This does not limit the reception of event-driven messages, e.g. in case of a remote accident, since every master has its own NFN and could forward those event-driven messages by the *WSM Scheduler* as described in Sec. 4. Additionally, our hybrid cluster approach is completely decentralized as it only relies on GPS, speed and direction data which is part of a BSM encapsulated in a WSM. Furthermore the cluster approach is simple to implement and reduces computational complexity, which is a prerequisite in order to meet real-time constraints.

According to the classification into FFN and NFN, the *Message Evaluation* building block in Fig. 2 is refined towards that classification. There are hierarchical evaluation stages responsible for FFN and NFN (see Fig. 3). Both comprise two main components: A *Data Fusion Unit* and a *Filter Unit*. The *Data Fusion Unit* collects necessary sensor data from other ECUs in the E/E network, mainly GPS, speed and radar data, incoming WSMs, and evaluation data of the external/internal communication monitoring components. The reasons for the hierarchical evaluation approach are to handle the complexity of the overall evaluation (divide and conquer), a better modularity and prefiltering of non-safety messages in high-density and critical traffic situations. Out of this data fusion, dedicated *acceptance policies* for FFN and NFN are specified. They are dynamically adapted mainly dependent current vehicle state and network utilization. The actual *Filter Unit* uses those policies for comparisons of data appearing in WSM in order to pass it into the nearfield evaluation stage, into the internal E/E network or discard it. In contrast to the *Farfield Evaluation Stage*, no WSMs are forwarded at the output of the *Nearfield Evaluation Stage*, but application specific data frames, which contain the most relevant application specific data of the nodes within the
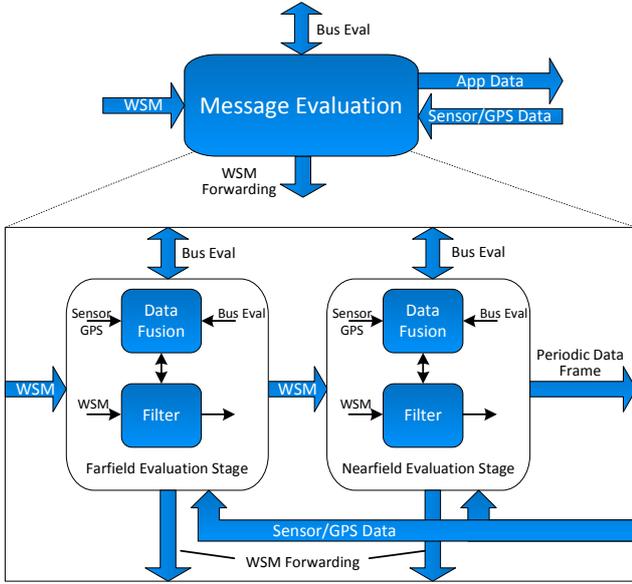
Figure 3: Refined Message Evaluation Building Block According to FFN and NFN Classification

NFN. Therewith, an application specific cost function is to be defined for each application, which weights the data of each node in the NFN. Finally these frames are periodically sent over the internal E/E network via the *E/E Architecture Interface* to the required safety application like CACC, since they depend on information updated regularly [6].

The hierarchical clustering and evaluation process is illustrated in Fig. 4. Because of the ad-hoc nature of vehicular networks, both FFN and NFN database must be updated regularly. It is updated every time a beacon or local sensor data from other ECUs is received. To limit database size and thus memory, FFN and NFN databases are deleted and refreshed respectively. Regarding FFN the whole database is deleted whereas in case of NFN only the nodes crossing the defined NF area are deleted. The necessary refresh cycles of FFN and NFN have been omitted for clarity in Fig. 4. In the following the cluster-based evaluation process is outlined in more detail.

4.1.1 FFN and NFN Cost Functions

The processes *Update Cluster Master* and *Update NF Node* in Fig. 4 are part of the *Data Fusion Units* and are triggered every time WSM or sensor/GPS data is received. They update the information position, driving direction, speed, priority and service identifier (PSID) of the current remote node entry v_j with received WSM data. Afterwards, the appropriate distance, relative position and driving direction between local vehicle v_i and v_j are calculated by means of local vehicle's sensor data. These values are then used for calculating the appropriate cost function of the current node in FFN and NFN respectively, weighting the relation between v_i and v_j. A cost function typically contains sub-functions evaluating both, mobility data and header data allowing cross-layer optimizations.

In case of **FFN cost function** a linear heuristic for long-term optimization and detection of hot-spots, i.e. congestion is used. The goal is to set the vehicle into a premature warn-
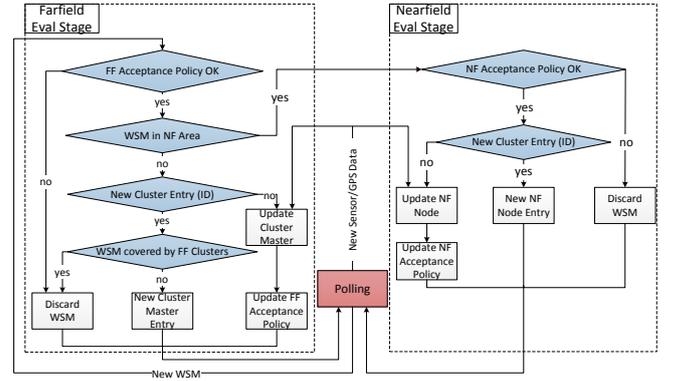

Figure 4: Hybrid Clustering and Hierarchical Evaluation Strategy

ing state aiming at only accepting safety-relevant messages. It is calculated between local vehicle v_i and each cluster master $v_j, \forall j \in FFN$ in the *Update Cluster Master* process and is comprising the following sub-functions:

- *compDriveDir()*: returns 1.0 if the driving direction of v_j is in range of v_i, otherwise 0.0. The range is a parameterizable constant.
- *psidEval()*: returns 1.0 if the received PSID in the WSM is contained in the set of cooperative warning messages defined in IEEE 1609.12, otherwise 0.0. Currently the following PSIDs are implemented: $0x0005$, $0x000C$, $0x2000$ and $0x8005$.
- *normDistance()*: returns values in $[0, 1]$, where a value of 1.0 represents the cluster master v_j with minimal distance to v_i.
- *compRelPos()*: returns 1.0 if the relative position of v_j is in range of relative position of v_i, otherwise 0.0. The relative position is represented in polar coordinates and the range is again a parameterizable constant.

Finally, the cost function $C_{i,j}^{FF}$ between v_i and v_j results in

$$
\begin{aligned}
C_{i,j}^{FF} = {} & a * compDriveDir() + b * psidEval() \\
& + c * normDistance() + d * compRelPos()
\end{aligned}
\tag{1}
$$

where *a-d* are weighting factors in $[0.0, 1.0]$ and $a+b+c+d = 1.0$. Thereby, $a = c = 0.1$ is chosen, because cluster master distance and driving direction of the whole cluster is only of less importance for hot-spot detection. In contrast, $b = 0.5$ is chosen since it is essential to detect event-driven warning messages and $d = 0.3$ to decide if an accident is basically possible. Finally, at the end of the *Update Cluster Master* process the maximum cost $C_{max}^{FF} = \max(C_{i,j}^{FF}), \forall j \in FFN$ is determined, triggering the driving state FSM which updates the acceptance policies.

In case of **NFN cost function** there has to exist a cost function for each application to be considered in order to forward the most relevant application specific data to the appropriate application via the *E/E Architecture Interface* (see Fig. 2). In case of safety applications, periodic frames have to be transmitted over the internal network. In this work we exemplarily specify a cost function for the CACC application as used in our case studies. According to [6] we consider the vehicle ahead with minimum distance and

moving in the same direction as most relevant for CACC. Therefore the cost function is similar to the FFN cost function. It is calculated for all neighboring vehicles $v_j \in NFN$ and is defined as follows:

$$C_{i,j}^{CACC} = compDriveDir() * normDistance() \\ * compRelPos()$$ (2)

At the end of the *Update NF Node* process the maximum cost $C_{max}^{CACC} = \max(C_{i,j}^{CACC}), \forall j \in NFN$ is determined. The corresponding latest data frame is periodically sent to the appropriate ECU via the *E/E Architecture Interface* at a parameterizable frame rate λ. For CACC this data frame consists of the distance to and speed of the vehicle ahead as well as the speed of the local vehicle. The frame rate is set to $\lambda_{CACC} = 0.1s$. Note that the ranges for the subfunctions $compDriveDir()$ and $compRelPos()$ have to be chosen much smaller than in case of FFN in order to select the correct vehicle directly ahead even in case of oncoming traffic. Additionally, if the costs get lower than a small delta, no vehicle ahead is assumed resulting in maximum acceleration of the CACC controlled vehicle until a maximum speed is reached or a vehicle ahead is detected again.

4.1.2 Acceptance Policies

Acceptance Polices are configured in the *Data Fusion Units*. They are a set of mobility and header data contained in BSM and WSM and are defined as $A_{NF} = \{prio, age, range\}$ and $A_{FF} = \{ID, psid, prio, age, range\}$ for nearfield and farfield respectively. They're dynamically parameterizable constants or ranges adjusted in the appropriate *Data Fusion Units* compared to the data of current processed WSM in the *Filter Unit* as depicted in Fig. 3. Thereby, *ID* is used to only accept beacons or WSMs dedicated for the local vehicle. *psid* is used for filtering a set of PSIDs. *prio* is used to only accept WSMs above or equal to a given priority ranging from 1 to 7. *age* limits the time interval elapsed since transmission and *range* defines an area range in terms of x, y, z-coordinates relative to the current position. In case of A_{NF} *range* defines the NF area as depicted in Fig. 4. There are dedicated acceptance policies for FFN and NFN allowing independent evaluation and filtering in each stage of possibly different strength. This is necessary e.g. in cases of high internal network utilization, where only WSMs with highest priority and nearest range could be accepted in NFN. Whereas FFN could accept lower prioritized messages used for cluster optimization but they are not forwarded to the nearfield stage. For all parameters there are two types of limits defined: *weak* and *strong*.

$\mathbf{A_{FF}}$ are mainly determined by the current vehicle state of the local vehicle represented by an FSM. However, *ID* is not set by the FSM. *ID* contains fixed values for the ID of the local vehicle and the broadcast ID. The FSM is triggered every time the *Update Cluster Master* process is finished or an update of local speed sensor is received. The FSM is depicted in Fig. 5. Basically, the FSM consists of a set of four states, namely *city, highway, danger, relaxation*. Transitions between states depend on the current speed retrieved by local sensors, on the current maximum cost of all cluster masters retrieved by *Update Cluster Master* process and on current wireless network utilization. *strong* acceptance policies are set in rather slow traffic situations below a certain speed limit $speed_{th}$, i.e. in state *city* or in potential dangerous

scenarios like congestion (state *danger*) crossing a cost limit C_{th}. *weak* acceptance policies are only set if there are no dangerous situations and if the speed crosses $speed_{th}$. The state *relaxation* serves as intermediate state which indicates relaxing traffic situation, but still has *strong* policies because a transition back to *danger* is still possible.
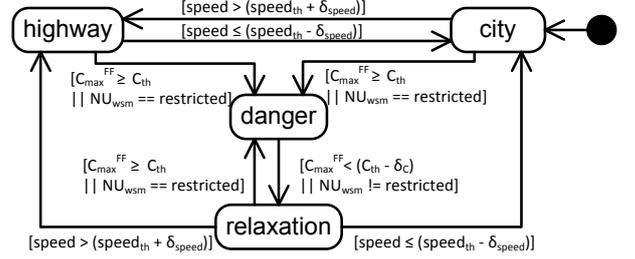

Figure 5: Driving State FSM responsible for adjusting A_{FF}

Because of the prefiltering of non-critical messages in potential dangerous situations in the farfield stage, control of $\mathbf{A_{NF}}$ must only depend on current speed and internal network utilization. Thereby, if NU_{ee} is in *restricted* mode, *strong* policies are applied and only safety related messages (e.g. BSMs and event-driven warnings) in NF range are passed in order to relax internal network utilization. To ensure that no potential safety related message is left unregarded, there is a minimum NF range, even when *strong* policies are active.

4.2 Network Monitoring and Evaluation

Network utilization $NU(\tau)$ is evaluated by a dedicated cost function for both WSM wireless channel used as well as for the internal E/E bus system. They influence the acceptance policies as described in Sec. 4.1.2. Unlike other approaches, e.g. [8] or [14], that exploit protocol specific MAC or PHY characteristics, we use a protocol independent metric to measure average network utilization. It is based on the average measured byte rate B_{wsm} and B_{ee} of actual transferred messages passing the monitoring units located at the appropriate interfaces (see Fig. 2 top) within a time interval T. These are normalized by the maximum nominal byte rate N_{wsm} and N_{ee} over the WSM channels and the E/E bus respectively. As WSMP transmits its nominal transmission rate per WSM, N_{wsm} is averaged in T, too. Overall, we assume only that the transmitted message size and the maximum nominal transmission rate is known, which is the case for WSM and for most of the protocols used in E/E architectures, e.g. CAN or LIN. The network utilization is additionally smoothed linearly [14] resulting in

$$NU_x(\tau) = 0.5 * NU_x(\tau - 1) + 0.5 * \frac{B_x}{N_x}$$ (3)

where x stands for indices *wsm* and *ee*. According to [14] we map the resulting values between 0.0 and 1.0 of NU to seven discrete states numbered from 1 to 7 indicating network utilization. If NU falls below a minimum value NU_{min}, the channel can be seen as free and is called *relaxed*. Above a maximum value NU_{max} it can be seen as congested and is called *restricted*. In case of WSM channel utilization this also indicates a potentially congested traffic scenario [14]. In between the states are called *active* and distanced by $(NU_{max} - NU_{min})/5$. Thereby, NU_{max} and NU_{min} could be selected by experimental or analytical data of the network. A

transition between states is only done if a certain threshold NU_{up} and NU_{down} is exceeded in order to increase state stability.

5. CASE STUDIES

In the following, a CACC application running on a future automotive E/E architecture has been chosen as an example to evaluate our approach. Thereby, two goals are pursued I) demonstration of a significant reduction of internal network traffic even in changing traffic situations and II) demonstration of accident-free CACC behavior when considering realistic processing and communication latencies.

5.1 Simulation Setup

The presented case studies are conducted with the co-simulation framework presented in [7]. It is based on Ptolemy II (PtII) [15] and on a simulation middleware based on the High-Level Architecture (HLA) [16] enabling distributed heterogeneous co-simulation. It allows investigation of interdependencies between inter/intra vehicle communication as well as computation overhead. Therewith the E/E architecture as shown in Fig. 1 of selected vehicles is modeled in PtII. The evaluated CACC application is running on the *Central ECU*. The *Wheel ECUs* requests the acceleration calculated by CACC controller and adjusts the desired speed. This intra-vehicle model is combined with a inter-vehicle model that is provided by Veins [17] v3.0 simulator. Veins is a vehicular network simulator that integrates the OMNET++ [18] network simulator with the SUMO [19] traffic simulator. They're communicating bidirectionally via TCP/IP. It allows detailed simulation of IEEE 802.11p/IEEE 1609.x DSRC PHY and MAC layers including multi-channel operation as well as node mobility. Veins simulates the surrounding vehicles and the inter-vehicle communication, whereas selected vehicles can be controlled remotely by PtII. This enables mobility data as well as beacons to be transferred between Veins and PtII.

Regarding synchronization a parameter can be set for each vehicle representing the time interval for updating messages exchanged with PtII. This comprises necessary sensor and GPS data as well as beacons that are exchanged. In the different scenarios following these parameters are set to $0.1s$, i.e. $10Hz$ as recommended update rate for CACC applications according to [6]. For DSRC PHY we use a transmit power of $20dBm$ in order to reach as many vehicles as possible. As well path loss and fading models are applied. Receiver sensitivity is set to $-89dBm$ and thermal noise to $-110dBm$. For DSRC MAC we use only the control channel (CCH) to communicate data at a rate of $6Mbps$ [20]. Network monitoring parameters are set to: $N_{ee} = 500kbps$, $NU_{min} = 0.15$, $NU_{max} = 0.40$, $NU_{up} = 0.005$ and $NU_{down} = 0.02$. The *strong* and *weak* acceptance policies of FFN and NFN are set according to Table 1. We prototypically chose one possible and feasible parameter set which could be used in real vehicles and traffic scenarios. Nevertheless, it's possible to set a various number of combinatorial possibilities of the parameters.

For the conducted scenarios we've constructed an urban traffic scenario in SUMO consisting of two adjacent lanes, where vehicles send BSMs at $10Hz$. We exemplarily chose 30 vehicles for simulation. Basically, an arbitrary number of vehicles can be simulated at the cost of simulation time. The leading vehicle is periodically approaching intersections

Table 1: Acceptance Policy Specification

	weak	strong
A_{FF}		
PSID	all	safety related
prio	all	$prio(BSM)$
range (x, y, z)	$300m$	$150m$
age	$1s$	$0.5s$
A_{NF}		
prio	all	$prio(BSM)$
range (x, y, z)	$300m$	$100m$
age	$0.5s$	$0.2s$

with a distance of $200m$ to each other. The mobility model of SUMO results in regular deceleration and acceleration cycles. A second follower vehicle is controlled remotely by PtII, whose speed is set with the value calculated by the CACC application. The rest of the vehicles are following the second one. The traffic simulation is executed with a fore-run of $10s$ in order to have an initial vehicle network for evaluation. The speed and position values of the leading vehicle required for CACC are transferred to PtII via the PHY/MAC model of Veins.

5.2 Scenario I

In this scenario the whole platoon described in Sec. 5.1 is performing a turnaround at the second intersection onto the opposite lane after about $40s$. In Fig. 6 and Fig. 7 the measurements of CACC are illustrated, showing the speed and position of the leading vehicle and the second, remotely controlled, local vehicle. As we can see, the CACC controlled vehicle is correctly following the leading vehicle with a certain safety distance. The deceleration and acceleration phases reflect save approaches towards the intersections. At the second intersection, the cost function $C_{own,ahead}^{CACC}$ indicates a maximum acceleration after the leading vehicle has turned onto the opposite lane (about $38s$) until the remote controlled vehicle itself reaches the intersection (about $42s$). Immediately after the turnaround of the remote controlled vehicle, the leading vehicle is detected again by the cost function and CACC is reacting properly even in case of oncoming traffic. Higher speeds are applied after the turnaround in order to decrease distance to the leader.

In Fig. 8 the number of overall received WSMs and the remaining number of messages after each evaluation stage within a time slot of $2s$ is illustrated. As we can see, farfield savings are small (or even zero) at the beginning, since most of the vehicles are in NF range. As time advances, more and more vehicles get into FFN. Compared to the overall number of received WSMs (dark blue bars) this results in a decreasing number of messages that remain after the farfield evaluation stage (blue bars). After the turnaround, vehicles of the oncoming traffic, which were nodes in the FFN before, join the NF range again. This results in an increasing number of messages that cannot be discarded in the farfield evaluation stage and are passed into the nearfield evaluation stage. Within the farfield stage up to $69,3\%$ and $37,0\%$ in average of all incoming messages are discarded. The constant number of 20 messages (light blue bars) show the periodic CACC frames generated in the nearfield evaluation stage. They correspond to the number of messages injected into the internal E/E network. This reflects a maximum message reduction of $96,6\%$ and $92,7\%$ in average and corresponds to a relative average reduction of the internal E/E network

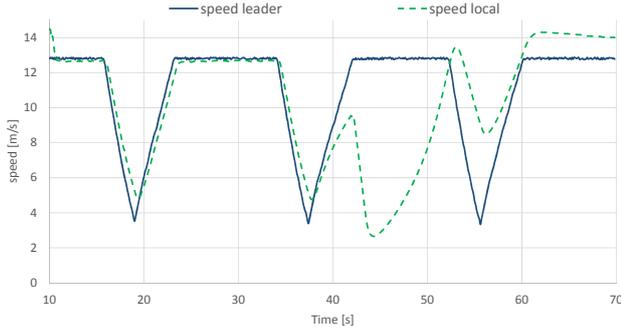load of $92,7\%$ (assuming a constant frame size).

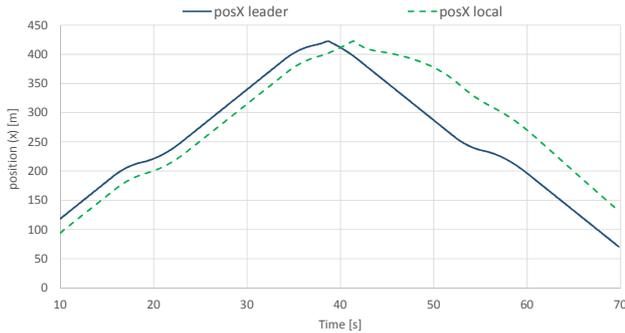**Figure 6: Speed of the leading vehicle and CACC controlled local vehicle**

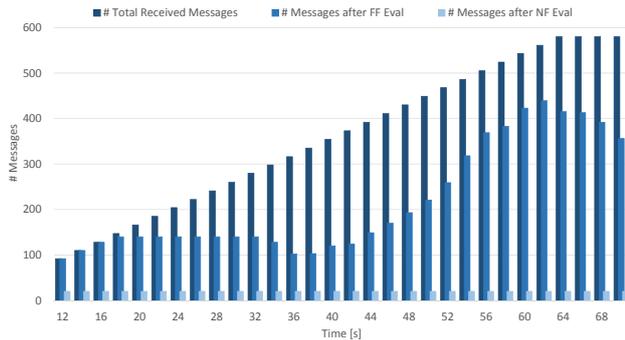**Figure 7: Position of the leading vehicle and CACC controlled local vehicle**

Figure 8: WSM Histogram of the Turnaround Traffic Scenario

5.3 Scenario II

In this scenario an accident is analyzed caused by the leading vehicle after $30s$. At this point in time, the leader sends out an event-driven warning message with PSID $0x8005$ indicating the accident. This results in *strong A_{FF}* acceptance policies. Furthermore, in this scenario we've especially considered internal E/E communication and processing latencies. We've generated synthetic internal network traffic in the E/E architecture model by so called abstract *aspects* provided by PtII. This results in *restricted NU_{ee}* state, only allowing BSMs and event-driven safety messages. According to [21] the total latency of a WSM should be much less than $100ms$. They define total latency as sensing of an event, communicating it to the target vehicle and taking action either by the driver or an active safety system. In [4] a "reception latency" of $< 10ms$ for e.g. pre-accident sensing events and 10-20ms

for BSMs is proposed. However it's not precisely defined what the starting and end point of the reception latency is. Since there are also latencies on the wireless channel (covered by Veins model) we therefore modeled a combined internal preprocessing, processing and communication reception latency of $10ms$. This serves as worst case for each received WSM until data reaches the actual destination processing unit, i.e. the CACC controller. Thereby, the preprocessing latency especially reflects necessary signature verifications of incoming WSMs. In addition, the CACC controller and the adjustment of the actual speed in the *Wheel ECU* causes processing latency. This latency modelling allows us to analyze how our approach can improve CACC behavior under realistic real-time latency constraints. In Fig. 9 the measured speeds of the leading and remote controlled vehicle are plotted for both cases with and without filtering.
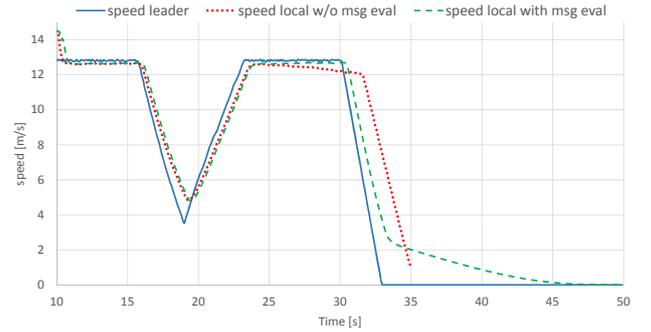

Figure 9: Speed of the leading vehicle and CACC controlled local vehicle with and without filtering

Even with only 30 vehicles and low urban speeds, constraints of $10ms$ latency of a WSM are not sufficient to meet the real-time requirements of CACC. This results in erroneous behavior of CACC. The WSMs of the participating vehicles cause a latency of the WSM of the leader containing the actual relevant data for CACC. This means the CACC controller processes potentially obsolete data. Without filtering this ends up in an accident reported by SUMO. This is indicated in Fig. 9, where the dotted red speed curve ends after about $35s$. It is obvious that in more large-scale scenarios with hundreds of vehicles and higher speeds, a latency of $10ms$ per WSM is by far not sufficient to meet the real-time constraints of CACC and potentially other safety applications. When applying our message evaluation strategy only messages that have passed the far- and nearfield evaluation stage induce the aforementioned latency chain (preprocessing, processing and network delay). Hereby, we assume that the evaluation stages induce negligible latencies compared to those mentioned previously. The dashed green speed curve in Fig. 9 shows that the reaction time of CACC with message evaluation and filtering is much smaller than that of the vehicle without filtering (dotted red curve). Thus a collision is prevented by asymptotically approaching the broke down leading vehicle.

6. SHORT DISCUSSION

The case studies show that a significant reduction of messages that internally need to be processed during run time can be achieved. Real-time requirements of a safety related CACC application can be met, even in the presence of both internal processing and communication latencies as well as critical traffic situations. Without filtering this could not be

accomplished. Hence we believe that our message evaluation approach can significantly improve reliability of cooperative safety applications. In addition, in combination with our co-simulation framework with detailed traffic and network models it enables early and reliable design space exploration of E/E architectures in V2X scenarios. Comparisons with results of previous approaches (see Sec. 3) is hard, since they consider a vehicle as a concentrated processing point, neglecting internal E/E architecture issues. However, a deeper analysis of various parameter settings especially of the acceptance policies can be addressed in future work (see Sec. 7). These could be additionally evaluated in more complex and large-scale traffic scenarios. Beyond that, because of the heavy incoming WSM data traffic at the farfield stage, the model serves as a foundation for performance and latency evaluations of future reconfigurable architectures including 3D FPGAs. Out of the model evaluations, design parameters for those architecture implementations are derived in order to meet the real-time requirements.

7. CONCLUSION AND FUTURE WORK
Within this work, a novel V2X message evaluation methodology based on a vehicle clustering strategy and monitoring capabilities of both vehicle and internal/wireless network utilization has been presented. The approach was modeled as part of a V2X ECU together with a CACC safety application as well as with abstract E/E architecture components. Two case studies with different traffic scenarios served as proof-of-concept of the approach. Future work comprises deeper analysis of the acceptance policies in more complex traffic scenarios. Also analysis of power consumption reduction of signature verification modules by our approach are planned by integrating appropriate hardware models into our co-simulation framework.

8. ACKNOWLEDGMENTS

9. REFERENCES
[1] C. Buckl *et al.*, "The software car: Building ict architectures for future electric vehicles," in *Electric Vehicle Conference (IEVC), 2012 IEEE International*, 2012, pp. 1–8.

[2] "IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments," *IEEE Std 802.11p*, Jun. 2010.

[3] "IEEE Standard for Wireless Access In Vehicular Environments (WAVE) - Architecture," *IEEE Std 1609.0*, Dec. 2013.

[4] "SAE Standard for Dedicated Short Range Communications (DSRC) Message Set Dictionary," *SAE Std J2735*, Nov. 2009.

[5] B. Glas *et al.*, "Car-to-car communication security on reconfigurable hardware," in *VTC Spring'09*, 2009.

[6] J. Ploeg *et al.*, "Design and experimental evaluation of cooperative adaptive cruise control," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, Oct 2011, pp. 260–265.

[7] C. Roth *et al.*, "A Simulation Tool Chain for Investigating Future V2X-based Automotive E/E Architectures," in *Embedded Real Time Software and Systems (ERTS²), 7th European Congress on*, Feb. 2014, pp. 1739–1748.

[8] C.-L. Huang *et al.*, "Adaptive intervehicle communication control for cooperative safety systems," *Network, IEEE*, vol. 24, no. 1, pp. 6–13, Jan 2010.

[9] Y. Feng *et al.*, "Adaptive beacon rate adjusting mechanism for safety communication in cooperative ieee 802.11p-3g vehicle-infrastructure systems," in *Communications (APCC), 2010 16th Asia-Pacific Conference on*, Oct 2010, pp. 441–446.

[10] R. Schwartz *et al.*, "Fair and adaptive data dissemination for traffic information systems," in *Vehicular Networking Conference (VNC), 2012 IEEE*, Nov 2012, pp. 1–8.

[11] ir. R.S. Schwartz *et al.*, "On the applicability of fair and adaptive data dissemination in traffic information systems," *Ad hoc networks*, vol. 13, no. B, pp. 428–443, 2014.

[12] H. Arkian, R. Atani, and S. Kamali, "Cluster-based traffic information generalization in vehicular ad-hoc networks," in *Telecommunications (IST), 2014 7th International Symposium on*, Sept 2014, pp. 1195–1200.

[13] C. Chen *et al.*, "A trust modeling framework for message propagation and evaluation in vanets," in *Information Technology Convergence and Services (ITCS), 2010 2nd International Conference on*, Aug 2010, pp. 1–8.

[14] "Intelligent Transport Systems (ITS); De-centralized Congestion Control Mechanisms for Intelligent Transport Systems operating in the 5 GHz range; Access layer part," *ETSI TS 102 687 V1.1.1*, Jul. 2011.

[15] J. Eker *et al.*, "Taming heterogeneity - the Ptolemy approach," *Proceedings of the IEEE*, vol. 91, no. 1, pp. 127–144, 2003.

[16] "IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)," *IEEE Std 1516.x-2010*, Aug. 2010.

[17] 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.

[18] A. Varga and R. Hornig, "An overview of the omnet++ simulation environment," ser. Simutools '08. ICST, Brussels, Belgium, Belgium: ICST, 2008, pp. 60:1–60:10.

[19] M. Behrisch *et al.*, "Sumo - simulation of urban mobility: An overview," in *SIMUL 2011, The Third International Conference on Advances in System Simulation*, Barcelona, Spain, 2011.

[20] D. Jiang, Q. Chen, and L. Delgrossi, "Optimal data rate selection for vehicle safety communications," in *Proceedings of the Fifth ACM International Workshop on VehiculAr Inter-NETworking*, ser. VANET '08. New York, NY, USA: ACM, 2008, pp. 30–38.

[21] Y. Li, "An overview of the dsrc/wave technology," in *Quality, Reliability, Security and Robustness in Heterogeneous Networks*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Springer Berlin Heidelberg, 2012, vol. 74, pp. 544–558.