Towards Realistic Evaluation of Collective Perception for Connected and Automated Driving

Georg Volk[1], Quentin Delooz[2,3], Florian A. Schiegg[5], Alexander von Bernuth[1], Andreas Festag[2,4] and Oliver Bringmann[1]

***Abstract*— Collective perception in Vehicle-to-Everything (V2X) communications allows vehicles to exchange pre-processed sensor data with other traffic participants. It is currently standardized by ETSI as a second generation V2X communication service. The use of collective perception as a communication service for future fully autonomous driving requires a thorough evaluation and validation. Most of the previous work on collective perception has considered large scale-simulations with a focus on communications. However, the perception pipeline used for collective perception is equally important and must not be neglected or over-simplified. Also, to study collective perception in detail, large-scale field testing is practically infeasible.**

In this paper we extend an existing simulation framework with a realistic model for V2X communications and sensor-data based processing delays. The result is a simulation framework that incorporates the entire collective perception pipeline, which enables to comprehensively study sensor-based perception. We demonstrate the capabilities of this enhanced framework by analyzing the delay of each component involved in the perception pipeline. This allows a detailed insight in end-to-end delays and the age of information within the environmental model of autonomous vehicles.

I. INTRODUCTION

Sensor data sharing using vehicle-to-everything (V2X) communications is an effective and low-cost solution to enhance the perception range of a vehicle's sensors. It is the basis for various advanced use cases for connected and automated driving. The European Telecommunications Standards Institute (ETSI) has completed a study item for sensor data sharing [1], named "Collective Perception" (CP) and is actively working on the standardization. CP is based on the periodic exchange of messages with the direct neighbours within the communication range. The study item implies important design decisions including the definition of the Collective Perception Message (CPM) and features of the communication protocol towards the future standard.

The CPM carries object lists, the vehicle's sensor configuration, and other data fields. Depending on the message frequency and the number of objects included, CPMs can considerably increase the channel load [2]. To study the trade-off between channel usage and information exchanged, large-scale simulations were performed with V2X communication as focus, such as in [3] and [4]. The reason behind that is that currently, large scale field tests are not viable because too costly and difficult to setup. Moreover, a simulator able to perform large scale simulations, i.e., with thousand of vehicles, and capable of performing realistic communication and perception in a reasonable amount of time, does no exist, yet.

Therefore, the simulations performed in [1] assumed idealistic perception, e.g., access to the information such as heading, speed, and acceleration without measurement errors for the objects detected, and considered perfect tracking. While this approach was suitable to establish the message generation and object inclusion rules of the CP service, the impact of realistic perception on the CP should be considered as well.

In this paper, we propose to extend the RESIST simulation framework introduced in [5] to analyze the CP service with a focus on the perception of objects. We break down the full detection and CP pipeline into components and analyze their respective latency and impact on the perception of objects. We describe the simulation parameters and collect results with well-known metrics in a highway scenario. We then compare the obtained results with the ones obtained in [1] and validate our communication model. Finally, we provide an analysis on how realistic perception will influence the CP service. For example, it should be considered that a major part of the current CP service design relies on the tracking and matching quality of the perceived and received objects. Indeed, with a poor tracking and matching quality, all redundancy and inclusion rules (see [1]) would become inefficient as the objects would be wrongly associated or considered as new. Moreover, the time required to perform these matching and tracking operations should be considered as it impacts the age of information of the objects managed in the vehicles' environmental model.

The remainder of this paper is organized as follows: After reviewing existing work in Section II, we provide technical background on the simulation framework in Section III. We analyze source of delays in Section IV and describe our simulation framework in detail in Section V. In Section VI we present and discuss our results. Finally, we conclude our paper in Section VII.

[1]University of Tübingen, Faculty of Science, Department of Computer Science, Embedded Systems Group {georg.volk, alexander.von-bernuth, oliver.bringmann}@uni-tuebingen.de

[2] CARISSMA – Research and Test Center for Vehicle Safety, Ingolstadt, Germany. {quentin.delooz, andreas.festag}@carissma.eu

[3] School of Information Technology, Halmstad University, Sweden

[4] Fraunhofer Institute for Transportation and Infrastructure Systems IVI.

[5] Robert Bosch GmbH, Corporate Research, Connected Mobility Systems, Hildesheim, Germany. florian.schiegg@de.bosch.com

II. RELATED WORK

Initial work on CP dates back to 2012 [6]. Ideas developed in [7] and others have led to standardization activities and to the publication of the ETSI study item TR 103 562 [1]. Several publications have reviewed the current design and elaborated on algorithms for CPM generation, such as [3], [4], [8]. These studies relied on the Artery [9] and the ns3-network simulators coupled to the Simulation of Urban MObility (SUMO) framework. Simulation configurations used in the study performed in [1] assumed that an object is perceived when in direct line of sight of a sensor. Additionally, the object processing delay was not considered and the tracking of objects was assumed to be perfect. In [10], the authors extended the Artery and SUMO frameworks with more realistic vehicle dynamics and probabilistic sensor models with the objective of enabling the generation of synthetic data for the CP service. In [11], the authors used the simulator CarMaker, real-world traffic racks and the Robot Operating System (ROS) as middleware. Other research items have considered different approaches than simulations. In [12], CP is experimented in a real setup. In [13] and [14] analytical models were used.

In this paper, we extend the RESIST simulator [5] and implement a perception-focused framework to analyze the CP service. In comparison to the other frameworks cited above, the presented simulator distinguishes itself by the usage of 3D simulation and a realistic camera sensor. The objective of this simulator is to provide the ability to study in detail how realistic perception influences the CP service. Potential applications for this simulator are the study of the redundancy control rules of the CP service (see [1]), especially the confidence and entropy based ones as they mainly rely on the perception of the objects, the analysis of the "look-ahead" mechanism presented in the same document, and the analysis of the parameters used in the inclusion rules of objects for CP. Our final goal is to perform studies combining results from the presented simulator with those of a dedicated network simulator (e.g., Artery).

III. BACKGROUND

In this section, the RESIST simulation framework as proposed in [5] is introduced. The simulation framework focuses on environmental conditions and includes a scenario-dependent simulation of the communication channel. In the RESIST simulator, the channel model is based on the distance between sending and receiving vehicles and incorporates obstructions in between. Important features that are relevant for communications, such as the Channel Busy Ratio (CBR) and the inclusion rules for the CP service as defined in [1], were missing. As these rules have a considerable impact on how many CPMs are generated and how many objects are included in each CPM, modifications were needed to align with the current standard development of CP. Additionally, the delay that occurs in the processing pipeline of CP was only considered as a whole. To study CP in more detail, improvements and measurements on the delay aspect in each processing step of CP need to be considered, especially as the delays have not been studied in depth yet. In the following sections we explain the modifications and improvements added to the simulator.

IV. DELAY ANALYSIS

The cooperative End-to-End (E2E) delay is the time between the detection of an object to its reception and processing by another vehicle. Figure 1 shows the decomposition of the cooperative E2E delay into three main components: the E2E delay of local perception, the V2X communication delay, and collective perception delay. Compared to other delay estimations such as proposed by [15], this approach allows for a more detailed and data dependent delay estimation as the delay is not obtained as a constant but is derived from each component in the processing pipeline. Further, as most components are highly dependant on environment (number of objects, number of cooperation partners, state of the communication channel, ...) and the deployed system (hardware, software, communication technology, ...) the main dependencies are discussed. The hardware was chosen such that the obtained values are comparable with those experienced in different projects such as [16] and [17].

Delay due to local perception: the local perception consists of four main components as illustrated in Figure 1: object detection, matching, tracking, and fusion. The object detection delay is defined as the difference between the capture time of the sensor data and the time when the sensor data processing is finished. The matching delay represents the time which is needed to associate detected objects with the currently present object tracks in the environmental model of the vehicle. Depending on the used tracking algorithm a separate matching might not be necessary and could be solved directly within the tracking algorithm itself [18]. The Tracking delay is defined by the time needed by the track management including the update or deletion of existing tracks and creation of new ones. The fusion of object tracks from different sensors is the last processing step resulting in the delay needed to fuse these tracks. If a vehicle is only equipped with one sensor a fusion is not needed.

Delay due to collective perception: the delay consists of three basic components: data alignment, matching, and track-to-track fusion. Objects contained in CPMs have an additional delay composed of the communication delay and the delay of asynchronous processing of the ego vehicle and other cooperative ones. Therefore, the first processing step in collective perception is the temporal alignment of data followed by the coordinate system alignment to the coordinate system of the ego vehicle [15]. As these two processing steps are the least computationally intensive in collective perception, these two delays are represented by the data alignment delay as shown in Figure 1. After data alignment, similar to local perception, the received objects have to be associated with the object tracks present in the ego-local environmental model. This results in the matching delay. After associating the ego-local object tracks with the cooperatively perceived ones the tracks have to be fused, which leads to a delay for this track-to-track fusion step.

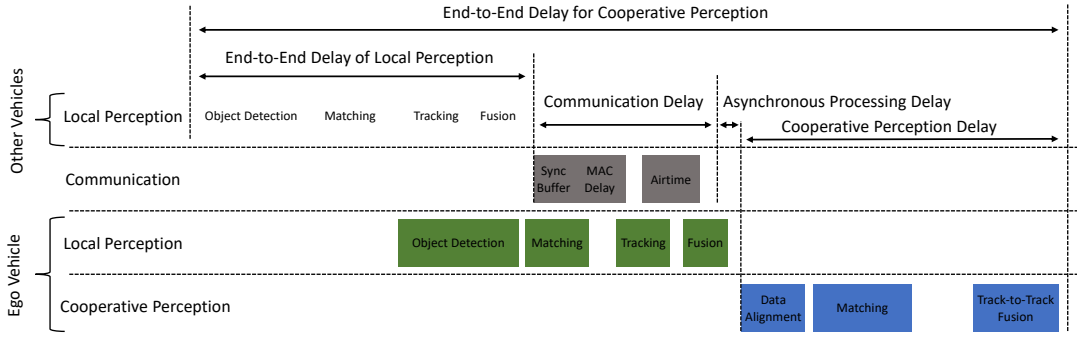

Fig. 1. Visualization of all delays in the collective perception pipeline. For visibility reasons, the proportions of visualized delays do not correspond to the proportions of observed delays from Table II.

Implementation of perception-based delays: the perception-based delays depend on the used processing algorithms and the number of perceived objects. For an adaptive use of the perception based delays, three possible delay representations for each processing component are introduced. An online delay d_o, a static offline delay d_s, and a dynamic offline delay d_d. Each detected object o within the simulation has a timestamp t and delay d. d is the accumulated delay of all processing steps of the pipeline.

The online delay d_o uses the actual delay per processing component. To reduce the influence of the processing overhead of the proposed framework only the actual processing time of the single components, i.e., matching will be measured. The online delay is strongly dependent on the used simulation hardware but has the advantage to represent the data dependencies in delays most accurately.

The static offline delay $d_s(\mu, \sigma)$ is defined by the mean μ and standard deviation σ per processing component. The actual delays are assessed by using a Gaussian distributed random variable defined by $N(\mu, \sigma)$. This allows to incorporate different processing delays as if the algorithms were executed on hardware which is actually used in autonomous vehicles. However, the static offline delay cannot represent the data dependent execution times, i.e., that matching a larger amount of detected objects takes longer than matching only few objects.

To combine the advantages of static offline delay and online delay, the dynamic delay is introduced to incorporate the data dependencies. The dynamic delay is defined as $d_d(n, \mu, \sigma)$, where, n is the sum of tracked objects in the environmental model and newly detected ones. μ and σ are the respective mean and standard deviations of the processing times per object in each processing component. Again, a Gaussian distributed random variable $N(\mu, \sigma)$ is used. This leads to:

$$d_d(n, \mu, \sigma) = \sum_{i=0}^{n} N(\mu, \sigma).$$

Delay due to V2X communications: the delay caused by V2X communications implementing the ITS-G5 protocol stack mainly depends on four different components: an introduced synchronization buffer, Decentralized Congestion Control (DCC), Enhanced Distributed Channel Access (EDCA) at the MAC layer, and the duration of transmission of a packet. Notice that within the scope of this paper, we did not consider security related and CPM decoding delays.

Before including objects in a newly generated CPM, the CP service should wait for the local perception pipeline to finish. This reduces the number of objects which will be included in the next CPM and therefore also reduces delays. Hence, a synchronization buffer is introduced. This buffer must be selected such that an object is postponed to the next CPM as rarely as possible. Not being included in the current CPM would increase the overall delay of an object by the CPM generation interval t_{CPM}. At the same time the synchronization buffer should be chosen as small as possible. Finding the optimal size of the synchronization buffer can be specified as optimization problem:

$$\min_{x \in \mathbb{R}^+} p_{\text{late}}(x) \cdot (t_{\text{CPM}} - \sigma \cdot x) + (1 - p_{\text{late}}(x)) \cdot \sigma \cdot x \quad (1)$$

x is the variable to optimize and defines the number of standard deviations of the local perception delay which will be used to specify the size of the synchronization buffer. p_{late} is the probability that an object is too late to be included in the current CPM. It is defined as $p_{\text{late}}(x) = 0.5 \cdot (1 - \text{erf}(\frac{x}{\sqrt{2}}))$, where erf is the error function evaluated at a given sigma x. σ represents the standard deviation of the local perception pipeline, t_{CPM} is the generation interval of CPMs which was set to $100\,\text{ms}$. Equation 1 is plotted in Figure 2. The optimal number of standard deviations x is 2.7 which results in a synchronization buffer of $2.67\,\text{ms}$.

The delay introduced by DCC depends on the current CBR and the priority of services generating messages on the same channel. Considering the adaptive DCC approach [19], if the channel load is low and only the CP service is using the channel, it is not expected that DCC will introduce additional delay [20]. In the case of the channel starting to be congested, DCC will start regulating the message transmission rate. In other words, DCC at the access layer will introduce a delay up to $1\,\text{s}$. If a message is delayed more than $1\,\text{s}$ by DCC, the message will be dropped before being transmitted.

Regarding the EDCA delay, it depends on the traffic class of the message and how many vehicles are within sensing range and trying to transmit on the channel. As shown in

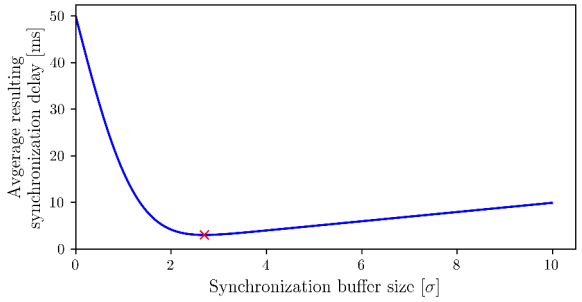

Fig. 2. Optimization function for finding the optimal number of standard deviations (σ) to specify the synchronization buffer. Red cross marks optimal synchronization buffer in σ.

[21], the EDCA delay should not be higher than a few ms in our simulations. However, it should be noted that in case of a congested channel, the EDCA delay would increase exponentially. It is worth mentioning that the medium access delay of IEEE 802.11p is generally considerably lower than that of Cellular-V2X in autonomous mode based on its Semi-Persistent Scheduling (SPS) mechanism. The latter is governed by the "selection window", which in turn is a function of the maximum tolerated latency and corresponds to the CPM generation period. SPS selects the most favorable resource within the selection window, thus leading to an additional delay of up to 100 ms.

The air time or equivalently the duration of a packet transmission in a channel depends on the packet size and the channel datarate. In our simulation, the channel datarate is $6\,\mathrm{Mbit\,s^{-1}}$. The packet air-time is determined by the following contributions: 0.04 ms (Physical layer) + (34 B (MAC header) + 2 B (LLC header) + 40 B (Geonetworking Single Hop Broadcast header) + 4 B (BTP header) + CPM size) / $6\,\mathrm{Mbit\,s^{-1}}$ (channel datarate). The CPM, as defined in [1], is composed of multiple containers: ITS PDU Header + Management Container + Originating Vehicle Container (45 B), sensor information (15 B, once per second), object container (48 B per object) and an optional freespace addendum. For example, for a CPM containing 12 objects (maximum number of objects included in a single CPM in our simulations), the message with all headers and the CPM has a size of 716 B. The resulting air-time is approximately 1 ms. It should be noted as well that the European standard [22] states that the maximum duration of a transmission on the channel shall not be higher than 4 ms.

V. SIMULATION FRAMEWORK

In this section we review our enhanced processing pipeline of the RESIST [5] simulation framework as illustrated in Figure 3. The main improvements compared to the original framework are the following: a thorough consideration of the MAC and physical layer, a per component based delay, the realization of ETSI CPM generation and inclusion rules, considering the CBR-dependent packet interference and collision effects to determine if a message could be correctly decoded or not, and improved detection and matching algorithms.

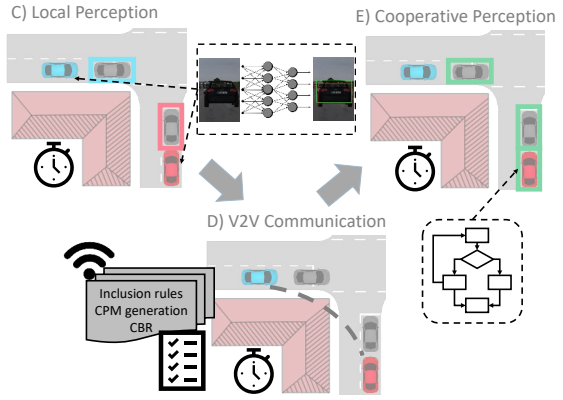

Fig. 3. Illustration of the improvements made to the original simulation framework. Main contributions are data-dependent delays for each processing step, realistic V2X communication modeling and enhanced perception algorithms.

A. V2X Simulation

For simulating the V2X communication channel, all packets to be transmitted by the cooperative vehicles within the simulation have to be identified. Afterwards the probability of detection for each packet has to be calculated according to the current CBR and distance between sender and receiver.

Each cooperative vehicle may generate a CPM every transmission interval t_{CPM}. The perceived objects, which will be included to a CPM are going to be restricted by the ETSI inclusion rules [1]. If no object is detected or none fulfills the inclusion rules no CPM will be generated. We incorporate the inclusion rules for position change r_{pos}, velocity change r_{vel}, change of course r_{course} and the maximum transmission time for vehicles $r_{t,\mathrm{veh}}$ and vulnerable road users (VRUs) $r_{t,\mathrm{vru}}$. The position change r_{pos} defines a minimum change of position of a perceived object before it is allowed to be included in a CPM again. Similarly r_{vel} and r_{course} define a minimum change in velocity or change in direction of driving. The dependence on the type of detected object $r_{t,\mathrm{veh}}$ defines the longest allowed time for a vehicle to not be included into a CPM respectively $r_{t,\mathrm{vru}}$ for VRUs. The track IDs id_{track} of the vehicle local perception systems are used to identify the same vehicle and apply the inclusion rules. If an object fulfills all inclusion rule checks it is included into the CPM. This check will be repeated for every cooperative vehicle which has perceived data in the current transmission interval of t_{CPM}.

The next step is to determine the size of all generated CPMs to calculate the channel load. The size of a CPM is determined by the static and dynamic information included. The static information s_{static} is composed of all headers as defined in IV and the Originating Vehicle Container. The dynamic information is composed of the sensor information $s_{\mathrm{sensor}}(c)$, which is transmitted once per second for each cooperative vehicle c transmitting a CPM and the number of detected objects. The size of CPM i to be transmitted in the transmission window Δt is defined as:

$$\mathrm{size}_{\mathrm{cpm}}(i) = s_{\mathrm{static}} + 48\,\mathrm{B} \cdot \mathrm{objs}(i), \tag{2}$$

where objs(i) is the number of objects contained in CPM i.

The size of all generated CPMs at within transmission window Δt is therefore:

$$\text{size}(\Delta t) = \sum_{i=0}^{N} \text{size}_{\text{cpm}}(i), \tag{3}$$

where N defines the number of CPMs transmitted within transmission window Δt. Subsequently, the generated data rate density, i.e., the data generated per time and area, can be obtained with:

$$dr = \frac{\text{size}(1\,\text{s}) + \sum s_{\text{sensor}}(c)}{1\,\text{s} \cdot A}, \tag{4}$$

where A is the transmission area of the scenario. If the simulation time is below 1 s, the current data rate will be scaled to 1 s. The corresponding channel load is then computed based on the analytical model of IEEE 802.11p developed and validated with Veins by Sepulcre et al. [23]. Finally, to spare computation resources, the packet delivery ratio (PDR) is obtained from a look-up-table generated based on the published source code of the previously mentioned model. The look-up-table links transmitter-receiver-distance, generated data rate density, CBR and PDR. For each transmitted CPM the reception is computed by generating random numbers and turning them into Boolean by using the reception probability as comparison threshold.

B. Perception Pipeline

The perception pipeline is based on the original publication [5]. For the vehicle local perception an image-based object detection is employed. Compared to the original publication YOLOv3 [24] was used instead of Faster R-CNN [25] for object detection as it provides higher object detection accuracy and 2.3 times faster processing speed on our test system. A L-shape based object detection provided by [26] is used to estimate the 3D position from the detected two dimensional bounding box together with the depth image provided by the used simulator [27].

After object detection the association of the newly detected objects to already tracked objects within the environmental model has to be done. The original framework used a nearest neighbor matching, which has the problem of not finding a global optimal solution. Hence, the Hungarian algorithm was applied for matching as it is able to find the global optimal solution for an assignment problem. Instead of using the Euclidean distance for the association of objects with already existing tracks, the Intersection-over-union (IoU) was used. However, the IoU cannot be used directly as association cost for the Hungarian algorithm as the higher the IoU the better the result. This is contrary to the goal of the Hungarian algorithm that tries to find the global optimum by minimizing the total cost. To solve this problem we defined the cost as:

$$C_{\text{IoU}} = \begin{cases} (1 - \text{IoU}(\text{track}, o)) \cdot m & \text{IoU}(\text{track}, o) > 0, \\ 1000 & \text{otherwise.} \end{cases} \tag{5}$$

Where IoU(track, o) defines the original IoU of an already managed track in the environmental model and a newly detected object o, and m is the maximum allowed association cost. After association of the objects, a Kalman Filter with constant velocity model is used for tracking. Next, the communication channel is simulated as described in Section V-A. For collective perception the communicated data has to be temporally aligned to the ego-vehicle's local perception. Afterwards a coordinate system transformation to the vehicle local coordinate system of the ego vehicle is performed. The cooperatively perceived vehicles will be associated with the Hungarian algorithm and get fused with a track-to-track fusion into the environmental model of the ego vehicle as already introduced in [5].

Fig. 4. Snapshot of the simulated highway scenario. The left image shows the birds-eye-view of the ego vehicle's environment and the right image presents the camera sensor view of the ego vehicle. The red and green boxes indicate the local and the collective perception, respectively.

TABLE I
SUMMARY OF THE SIMULATION PARAMETERS

Parameter	**Values**
CPU	Intel-Core i7-7700K
RAM	32 GByte
GPU	RTX 2080Ti
Protocol stack	ITS-G5
CPM generation interval	[100 ms, 1 s]
Inclusion rule parameterization:	
r_{pos}	4 m
r_{vel}	$0.5\,\text{m}\,\text{s}^{-1}$
r_{course}	0.07 rad
$r_{t,\text{veh}}$	1 s
$r_{t,\text{vru}}$	0.5 s
Scenario	Highway (6 lanes) from [5]
Density of vehicles	100 veh/km (50 veh/km per driving direction)
Vehicle sensor equipment	1 camera, 30° FoV, placed centrally on the windshield
Number of vehicles in scenario	20
Length of simulation	25 s
Equipment Rate (ER)	{10, 25, 50, 75, 100} %
Number of repetitions	1,000

VI. EVALUATION

For evaluation a six lane highway scenario as depicted in Figure 4 was used. Table I shows the parameters used in the evaluation of the simulation framework explained in IV. To evaluate the environment perception only the three lanes in the direction of travel were considered. This reduces errors

TABLE II

OBSERVED DELAYS FOR A SINGLE SIMULATION RUN.

Delay	Mean [ms]	Stdev [ms]
Object detection	14.48	0.85
Local matching	0.97	0.51
Local tracking	0.04	0.02
Synchronization buffer	2.67	0.00
Channel access	1.50	0.50
Airtime	0.54	0.07
Data alignment	0.02	0.02
Cooperative matching	4.79	3.23
Cooperative track fusion	0.12	0.10

due to objects exiting the sensors FOV as only a forward facing camera per cooperative perceiving vehicle was used. For communication the ITS-G5 protocol stack was used. The results presented in the following of this section are collected at a determined vehicle in the simulation scenario.

A. Delays

To evaluate the perception pipeline delays we first measured the online delays as introduced in Section IV for the whole simulation run. The mean and standard deviations for the online delays are shown in Table II. It has to be noted that the delays for the local fusion component are not listed as our simulation only includes a single sensor. Therefore, no fusion for local perception has to be performed. The observed μ and σ values can be used for static offline simulation. Also, the measured processing times are strongly dependent on the hardware used for running these tests. Therefore, we list the hardware specifications in Table I. We have observed that aside of object detection and communication, the delay for matching was the most dominant part of the perception pipeline with a mean of $0.97\,\mathrm{ms}$ for local matching and a mean of $4.79\,\mathrm{ms}$ for cooperative matching. For further investigations, we considered the online delays only as they represent the most realistic representation of data dependencies in the processing pipeline. By using the dynamic offline simulation the obtained results could be easily transferred to a different hardware by measuring execution times on these components and identifying the hardware dependant delays per object as introduced in Section IV.

The synchronization buffer for the object inclusion is the largest of the delays caused by the V2X protocol stack, followed by EDCA. The maximum observed CBR was not higher than 0.3 (see VI-B) which is not enough for DCC to start actively restricting the CPM generation rate and the maximum observed packet air-time was 1 ms.

B. Metrics

Matching: on average, the number of objects included per CPM is around 5, independent of the equipment rate (ER). The average miss-matching of objects is $9.6\,\%$ which results in around 3.5 miss-matched objects per vehicle per second. A miss-match happens when a detected object is not associated to the correct object track or when a new track is created while the object is already tracked. Wrong object association

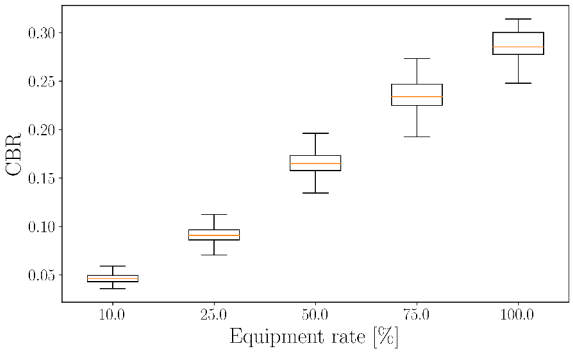

Fig. 5. CBR for different equipment rates. A significant increase of CBR with increased equipment rates was observed.

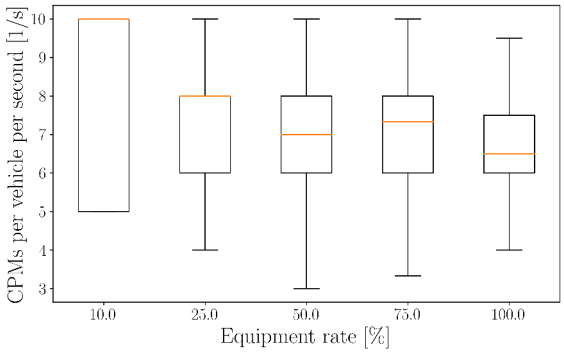

Fig. 6. CPMs generated per vehicle per second for different equipment rates. The number of generated CPMs per vehicle is almost constant.

is the consequence of bad sensor measurements in complex vehicle configurations. Losing track of objects may be due to either bad measurements or when the object moves out of the detection range of the sensor. In both cases, miss-matching impacts the CP inclusion rules. Indeed, a miss-matched object can be considered as newly detected and will be included in the next CPM generated. This artificially increases the number of detected objects and potentially creates more CPMs.

Channel Busy Ratio is a time-dependent value between zero and one representing the fraction of time that a single radio channel is sensed busy. If the CBR = 1, it means that the channel is sensed fully loaded. Figure 5 shows the obtained results. At an Equipment Rate (ER) of $100\,\%$, the maximum CBR is at around 0.3. When comparing to results obtained in the ETSI TR about CP [1], the magnitude of the CBR matches with the CBR obtained in the TR analysis (CBR between 0.3 to 0.36 with a vehicle density of 120 veh/km in an highway scenario).

CPM rate is the number of CPMs transmitted per second per vehicle in average. Figure 6 shows the obtained results. Each vehicle transmits about 7 CPMs per second. Only with low ER the median of CPMs per second is slightly higher. This is due to the structure of the data. The mean however for $10\,\%$ ER is at 8.5 CPMs per second. The observed median CPMs per second show a slight decrease with higher ER.

Time between updates & redundancy: Figure 8 shows the average time between two updates of the same object on the communication channel. The maximum time until the next update can reach up to 9.5 s. With 100 % ER the maximum update time drops significantly to 0.4 s. This clearly shows the benefit of collective perception with high ER rates. Furthermore, the mean between two updates decreases from 0.243 s for an ER of 10 % to just 0.056 s for 100 % ER. While it is important to receive periodic updates about objects, unnecessary communications should be avoided. Figure 7 shows the number of updates received per second in average of a same object. The higher the ER, the higher the redundancy of information. Some objects can be received more than 20 times per second. To better control the information exchanged on the channel, some of the proposed redundancy mitigation rules explained in [1] should be used and will investigated in further work.

End-to-End (E2E) delay is explained in Section IV. It is the difference between the time of the sensor measurement the object originated from and the time when it was merged into the environmental model. Figure 9 and Figure 10 show the local E2E delay and the cooperative E2E delay respectively. Not only the cooperative E2E delay increases with higher ER but also the local E2E delay increases. This is caused by the higher number of objects being present in the environmental model, resulting in more computational load for matching and track updates. The mean of the local E2E delay increases by 2.21 ms from 15.05 ms for ego perception only to 17.27 ms for an ER of 100 %. For the cooperative E2E delay an increase of 14.10 ms was observed. The mean E2E delay increased from 118.59 ms from 10 % ER to 132.70 ms for 100 % ER. The increase in cooperative E2E delay comes partially from medium access times in the communication channel and from the computationally more intense perception pipeline including matching and tracking.

Age of Information represents the average age of information present in the environmental model of the ego vehicle. The age of information is the difference between the timestamp of the last perceived sensor measurement of an object and the current time. This metric was assessed for all present objects in the environmental model every 100 ms. This metric includes locally and cooperatively perceived objects. The observed ages of information are shown in Figure 11. The biggest increase in age of information is caused by moving from ego perception only to collective perception. Making use of collective perception with 10 % ER increased the mean age of information from 135.17 ms to 207.19 ms. Increasing the ER further to 100 % increases the average age of information to 268.99 ms. The strongly visible clusters every 100 ms are resulting from the sensor frequency of 10 Hz.

VII. CONCLUSIONS & FUTURE WORKS

In this paper we have presented a comprehensive framework for collective perception, which incorporates realistic delays for every processing step within a perception pipeline. We performed simulations using an highway scenario and

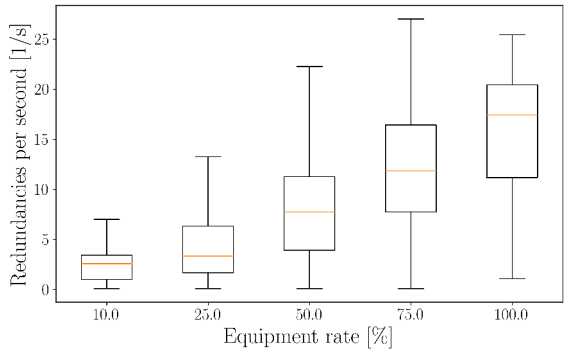

Fig. 7. Object redundancies per second for communicated objects for different equipment rates.

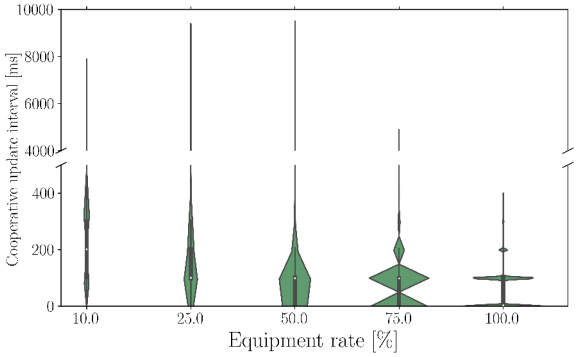

Fig. 8. Time in milliseconds between two updates of the same object on the communication channel for different equipment rates.

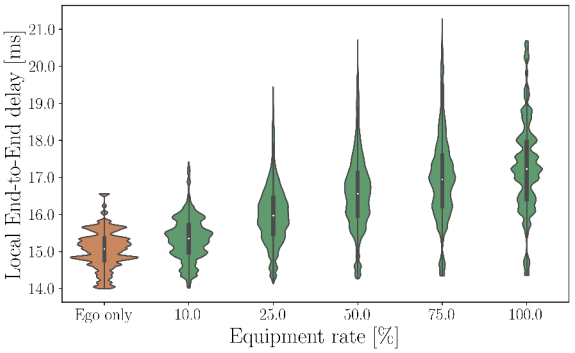

Fig. 9. End-to-End delay for vehicle-local perception. The observed delay increases with increasing equipment rates.

showed that not only the cooperative E2E delay increases with higher ER but also the local E2E delay increases. This comes due to the higher number of objects being present in the environmental model, resulting in more computational load for matching and track updates. In addition, the increased computational effort in the processing pipeline due to the larger number of objects can be considered. We have shown how different values of the equipment rate affect the End-to-End delay of the collective perception as well as the local perception. Due to the higher number of processed and transmitted objects with a growing equipment rate, the

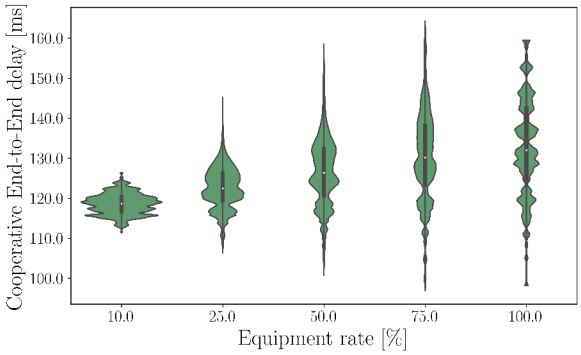

Fig. 10. Cooperative End-to-End delay. A significant increase of delay was observed with higher equipment rates

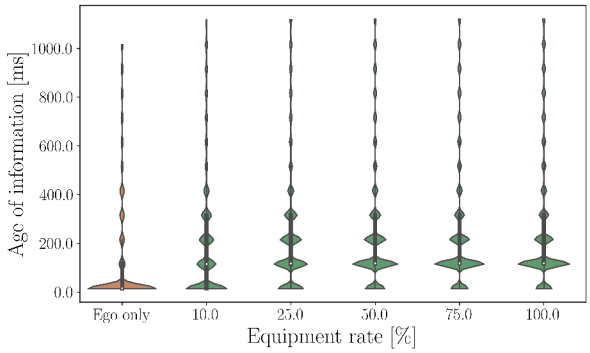

Fig. 11. Age of information in the environmental model of the ego vehicle including locally and cooperatively perceived objects.

delays for collective perception but also for local perceptions increase. Furthermore, we introduced an optimal synchronization buffer to improve collective perception delays such that the overall delay gets reduced and locally perceived objects will rarely be postponed to the next CPM generation interval.

In future work we will add a multi-sensor setup for the simulated vehicles and use improved heterogeneous perception algorithms. This allows an even better reflection of reality and a more realistic evaluation of CP services. Additionally, we will compare the simulation by using Artery with the same scenario and consider security for the communication channel. The presented simulator represents an important contribution for standardization at ETSI to support the numerous study groups for the CP services.

ACKNOWLEDGMENT

This work was supported by the German Science Foundation (DFG) within the priority program Cooperatively Interacting Automobiles (CoInCar) (SPP 1835).

REFERENCES

[1] ETSI, "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Analysis of the Collective Perception Service (CPS); Release 2," Dec. 2019, ETSI TR 103 562 V2.1.1.

[2] Q. Delooz and A. Festag, "Network load adaptation for collective perception in V2X communications," in *IEEE ICCVE*, Nov. 2019.

[3] K. Garlichs, H. Günther, and L. C. Wolf, "Generation rules for the collective perception service," in *IEEE VNC*, Dec. 2019.

[4] G. Thandavarayan, M. Sepulcre, and J. Gozalvez, "Analysis of message generation rules for collective perception in connected and automated driving," in *IEEE IV Symposium*, 2019, pp. 134–139.

[5] G. Volk, A. von Bernuth, and O. Bringmann, "Environment-aware development of robust vision-based cooperative perception systems," in *IEEE IV Symposium*, 2019, pp. 126–133.

[6] A. Rauch, F. Klanner, R. Rasshofer, and K. Dietmayer, "Car2X-based perception in a high-level fusion architecture for cooperative perception systems," in *IEEE IV Symposium*, Jun. 2012, pp. 270–275.

[7] H. Günther, O. Trauer, and L. Wolf, "The potential of collective perception in vehicular ad-hoc networks," in *ITST*, 2015.

[8] Q. Delooz, A. Festag, and A. Vinel, "Revisiting message generation strategies for collective perception in connected and automated driving," in *VEHICULAR 2020*, Oct. 2020.

[9] R. Riebl, H. Günther, C. Facchi, and L. Wolf, "Artery: Extending Veins for VANET applications," in *MT-ITS 2015*, Jun. 2015, pp. 450–456.

[10] C. Allig and G. Wanielik, "Extending the vehicular network simulator Artery in order to generate synthetic data for collective perception," *Advances in Radio Science*, vol. 17, pp. 189–196, 2019. [Online]. Available: https://ars.copernicus.org/articles/17/189/2019/

[11] F. A. Schiegg, J. Krost, S. Jesenski, and J. Frye, "A novel simulation framework for the design and testing of advanced driver assistance systems," in *IEEE VTC-Fall*, 2019, pp. 1–6.

[12] M. Shan *et al.*, "Demonstrations of cooperative perception: Safety and robustness in connected and automated vehicle operations," *Sensors*, vol. 21, no. 1, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/1/200

[13] F. A. Schiegg, D. Bischoff, J. R. Krost, and I. Llatser, "Analytical performance evaluation of the collective perception service in IEEE 802.11p networks," in *IEEE WCNC*, 2020, pp. 1–6.

[14] H. Huang *et al.*, "Data redundancy mitigation in V2X based collective perceptions," *IEEE Access*, vol. 8, pp. 13 405–13 418, 2020.

[15] A. Rauch, "Entwicklung von Methoden für die fahrzeugübergreifende Umfelderfassung," Dissertation, Ulm University, 2016, https://bit.ly/32yucZD.

[16] I. Llatser, T. Michalke, M. Dolgov, F. Wildschütte, and H. Fuchs, "Cooperative automated driving use cases for 5g v2x communication," in *2019 IEEE 2nd 5G World Forum (5GWF)*, 2019, pp. 120–125.

[17] F. A. Schiegg, I. Llatser, H. Tchouankem, F. Wildschütte, and F. Hofmann, "Removing blind spots: infrastructure-assisted collective perception," in *International VDI-Congress Electronics in Vehicles (ELIV'21)*, 2021.

[18] K. Granström, M. Baum, and S. Reuter, "Extended object tracking: Introduction, overview, and applications," *Journal of Advances in Information fusion*, vol. 12, no. 2, pp. 139–174, Dec. 2017.

[19] ETSI, "Intelligent Transport Systems (ITS); Decentralized Congestion Control Mechanisms for Intelligent Transport Systems operating in the 5 GHz range; Access layer part," Apr. 2018, ETSI TS 102 687 V1.2.1.

[20] Q. Delooz, R. Riebl, A. Festag, and A. Vinel, "Design and performance of congestion-aware collective perception," in *IEEE VNC*, 2020, p. 8.

[21] ETSI, "Intelligent Transport Systems (ITS); STDMA recommended parameters and settings for cooperative ITS; Access Layer Part," Jan. 2012, ETSI TR 102 861 V1.1.1.

[22] ETSI, "Intelligent Transport Systems (ITS); Radiocommunications equipment operating in the 5 855 MHz to 5 925 MHz frequency band; Harmonised Standard covering the essential requirements of article 3.2 of Directive 2014/53/EU," Feb. 2017, ETSI EN 302 571 V2.1.1.

[23] M. Sepulcre, M. Gonzalez-Martín, J. Gozalvez, and R. Molina-Masegosa, "Analytical models of the performance of IEEE 802.11p vehicle to vehicle communications," *ArXiv:2104.07923 [cs.NI]*, 2021, https://arxiv.org/abs/2104.07923.

[24] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *CoRR, abs/1804.02767*, 2018, http://arxiv.org/abs/1804.02767.

[25] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[26] X. Zhang, W. Xu, C. Dong, and J. M. Dolan, "Efficient L-shape fitting for vehicle detection using laser scanners," in *IEEE IV Symposium*, Jun. 2017, pp. 54–59.

[27] [Online]. Available: https://vires.mscsoftware.com/