Paper published at the 8th IEEE International Conference on
Connected Vehicles and Expo (ICCVE) in November 2019

# Decentralized Dynamic Platooning Architecture with V2V Communication Tested in Omnet++

Tobias Renzler
*Institute of Automation and Control*
*Graz University of Technology*
A-8010 Graz, Austria
Email: tobias.renzler@tugraz.at

Michael Stolz
*Institute of Automation and Control*
*Graz University of Technology*
*Virtual Vehicle Research Center*
A-8010 Graz, Austria
Email: michael.stolz@tugraz.at

Daniel Watzenig
*Institute of Automation and Control*
*Graz University of Technology*
*Virtual Vehicle Research Center*
A-8010 Graz, Austria
Email: daniel.watzenig@tugraz.at

***Abstract*—An open question in platooning is the appropriate selection of the inter-vehicle distance: shorter distances reduce energy, but increase network demands, which have to be satisfied to guarantee safety. Even once determined appropriately, inter-vehicle distances need to adapt for varying traffic scenarios and heterogeneous platoons. Therefore, we introduce a new architecture that puts dynamic distances into effect and manages platoons in a decentralized way to maintain safety and robustness by considering network reliability, environmental, and vehicular constraints. Smooth trajectory planning allows energy efficient and comfortable adaptation of distances between vehicles. The presented architecture is tested using Omnet++ and SUMO in a highway entry and an emergency braking scenario.**

***Index Terms*—C-ITS, CACC, CAV, platooning, platoon management, IEEE 802.11p, Omnet++, SUMO**

## I. INTRODUCTION

During the last years, platooning became one of the main topics in the research of Intelligent Transport System (ITS) applications. Platooning unites management, control, and communication aspects to increase safety, fuel economy, traffic throughput, and comfort. A platoon consists of several vehicles sharing acceleration, velocity, position, and sometimes also steering information. All vehicles except the leader use Cooperative Adaptive Cruise Control (CACC) to maintain a predefined distance to the preceding vehicle.

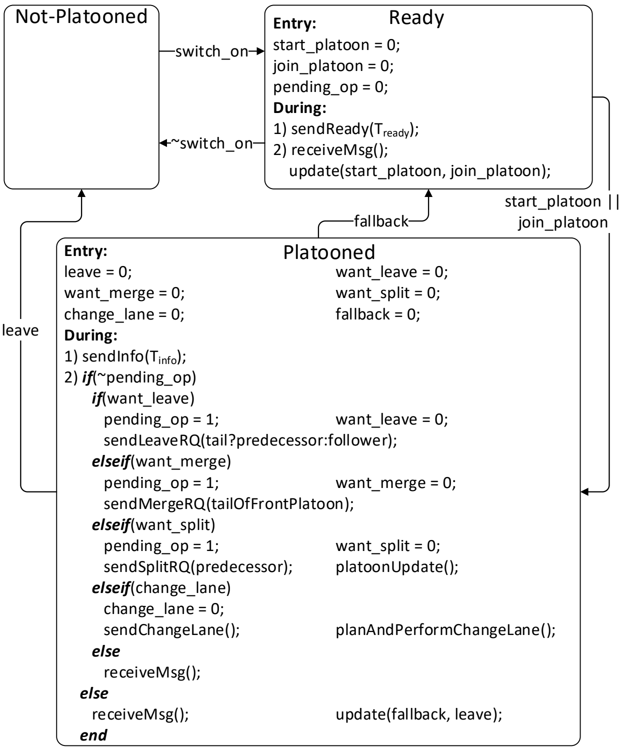
Due to aerodynamic drag reduction, platooning lowers energy consumption between $5\,\%$ and $25\,\%$. Experiments have proven, shorter distances lead to a smaller aerodynamic drag, and thus, to lower energy consumption up to an inter-vehicle distance of about $10\,\mathrm{m}$. For smaller distances no substantial reduction in energy consumption could be observed and cooling problems, especially for trucks, may arise [1].

The compliance of the desired inter-vehicle distance is achieved by the used control policy reaching from linear approaches such as constant spacing, constant time headway, and constant time delay to non-linear solutions. An overview on state of the art is given in [2]. Most control strategies are focusing on homogeneous platoons, i.e. vehicles of the same type, with the same load, same capabilities and cover only few low complexity scenarios (highways). Contrary to that, heterogeneous platooning control concepts target more realistic applications [3]: velocity, load, and position of a vehicle within a platoon influence the energy consumption [4]. Further, vehicle limitations, e.g., limited acceleration due to heavy load, and external circumstances such as road gradients have to be considered [5]. This raises the question of an appropriate inter-vehicle distance that guarantees safety while accounting for a dynamic environment.

Although a lot of research is done in control of inter-vehicle distances, platooning is much more than just distance control [6]. To ensure safety while using short inter-vehicle distances down to $10\,\mathrm{m}$, wireless communication technologies have to be used. Sharing data among vehicles of a platoon requires a highly reliable, short end to end delay, and data-fresh (i.e., short transmission intervals to keep data permanently updated) network. Nowadays, two different technologies are used in the field of Vehicle-to-Vehicle (V2V) communication: ad hoc (VANET) and cellular networks. There exist several comparisons of the performance in platooning scenarios of both types of networks, considering IEEE 802.11p and 3GPP long-term evolution (LTE) [7] [8]: both technologies suffer under mutual interference in high density networks resulting in varying network quality. This limits the amount of vehicles within a single network (one platoon), but also the number of multiple networks (multiple platoons) in interference range [9]. Packet size and latency further aggrevate network communication [10]. *Thus, to maintain a platoon it is crucial to send as little data as possible but as much as necessary.* The current standard for spreading vehicular information in 802.11p networks (ETSI EN 302 637-2 V1.3.2) introduced dynamic message generation in order to relief the network and still be able to react fast enough on a rapidly changing environment (e.g., emergency braking of front vehicle). A detailed investigation of those messages, called cooperative awareness messages (CAM), can be found in [11].

In this contribution we tackle the questions: *How to manage a platoon with low overhead?* and *How to determine appropriate vehicle specific inter-vehicle distances in the presence of heterogeneous platoons, varying network quality, and a highly dynamic environment?*

We first present a low overhead, decentralized platoon management architecture in Section II covering all platoon maneuvers from formation to dissolution. Afterwards, Section III shows an approach how each vehicle determines its dynamic distance fulfilling predefined safety requirements. In



```
Entry:
start_platoon = 0;
join_platoon = 0;
pending_op = 0;
During:
1) sendReady(T_ready);
2) receiveMsg();
   update(start_platoon, join_platoon);
```

```
Entry:
leave = 0;                          want_leave = 0;
want_merge = 0;                     want_split = 0;
change_lane = 0;                    fallback = 0;
During:
1) sendInfo(T_info);
2) if(~pending_op)
     if(want_leave)
       pending_op = 1;              want_leave = 0;
       sendLeaveRQ(tail?predecessor:follower);
     elseif(want_merge)
       pending_op = 1;              want_merge = 0;
       sendMergeRQ(tailOfFrontPlatoon);
     elseif(want_split)
       pending_op = 1;              want_split = 0;
       sendSplitRQ(predecessor);    platoonUpdate();
     elseif(change_lane)
       change_lane = 0;
       sendChangeLane();            planAndPerformChangeLane();
     else
       receiveMsg();
   else
     receiveMsg();                  update(fallback, leave);
   end
```

Fig. 1. State machine for platoon management consisting of states *Not-Platooned*, *Ready*, and *Platooned*.

Section IV, both platoon management and dynamic distance calculation are tested using Omnet++ and Simulation of Urban Mobility (SUMO). Finally, we discuss the presented architecture given two different traffic scenarios, operating in an 802.11p network and conclude the work in Section V, giving an outlook on future investigations.

## II. PLATOON MANAGEMENT

By platoon management we understand creation, maintenance, and dissolution of a platoon. For the behavioral execution of the presented platoon management we propose a state machine approach, composed by three distinct states (see Fig. 1): first, a vehicle is *Not-Platooned*, meaning the vehicle is under human control and also not willing to platoon at all. Second, a vehicle is *Ready* once the driver signalized the readiness to platoon, and third, a vehicle can be in state *Platooned*. The latter state is reached once a vehicle successfully created a platoon or joined an existing platoon.

Vehicles in state *Ready* or *Platooned* may perform different platoon maneuvers: form, dissolve, join [12], merge, split, change lane, and leave. The duration to complete maneuvers varies and is situation dependent [13]. Their processing may be based on a centralized or a decentralized architecture. In a centralized architecture one vehicle (usually the leader) governs the platoon [14]. Such an approach benefits from simplicity, as one vehicle decides if and how the structure of a platoon is changed. The drawback is the sequentialization of maneuvers that affect only part of the platoon: imagine a platoon of 10 vehicles driving on a highway, where the second and the last vehicle want to leave, to take the next exit. In a centralized approach, even though both leaving vehicles do not influence the role of the governing leader, both have to ask for permission to leave the platoon. As the governor can process only one request per time, other requests are delayed. Furthermore, due to the high spatial distance between the last vehicle and the leader, packets may need to be forwarded or resent due to packet loss. This causes communication delays, unnecessarily high network traffic, and introduces scaling restrictions for long platoons.

A more natural approach is a decentralized platoon management architecture as presented in this paper minimizing packet forwarding and keeping small communication distance. Alam et al. [15] outline the advantages of a decentralized controller for platooning making the system less sensitive to communication delays and physical limitations such as radio range. This holds also for the platoon management. In a decentralized architecture, a request for a specific maneuver goes to the vehicle that is directly affected by the maneuver and is also processed by the same vehicle. In case of the previous example, the second vehicle sends its request to its follower, as that vehicle receives a new predecessor. Once the last vehicle of the platoon leaves, it notifies its predecessor such that the platoon structure can be updated.

The message flow for all necessary maneuvers is illustrated in Fig. 2. *Ready* and *Info* messages are periodically broadcast with a period of $T_{\text{ready}}$ and $T_{\text{info}}$ respectively, and do not have a specific recipient, while all other messages are event triggered. In event triggered send procedures the recipient is named in parenthesis, where TX means it is a reply. Only addressed vehicles are expected to react on addressed messages, while messages with no recipient may be processed by any vehicle.

Each vehicle holds a topology of the current platoon, the so called platoon map. Information about the layout of the map and its last modification is spread using *Info* messages. Due to decentralization, the platoon map of vehicles may differ. This is unproblematic, as long one vehicle does not process two maneuvers at the same time. To avoid this, the variable *pending_op* is used for each vehicle. It is set/reset after starting/completing a maneuver. Vehicles may update their own map intertwiningly once an *Info* message is received: each vehicle knows its responsibility for certain maneuvers, allowing overwriting platoon map sections accordingly. Maneuvers and responsible vehicles are described as follows:

**Form:** Once a vehicle A is *Ready* it sends periodically messages of type *Ready*. Vehicle B, in state *Ready* receiving the message, checks if platooning conditions are met (e.g., B in front of A), creates a new platoon, becomes its leader, and invites A. The latter can accept/reject the invite (see *join*). If rejected, B falls back to state *Ready*.

**Dissolve:** It may occur that a whole platoon has to be dissolved. That may be due to specific traffic scenarios and may be requested by infrastructure. A *dissolve* procedure boils down to a sequence of *leave* procedures starting from the head of the platoon. This step-wise dissolution of the platoon offers a safe procedure, as not all drivers need to take control over

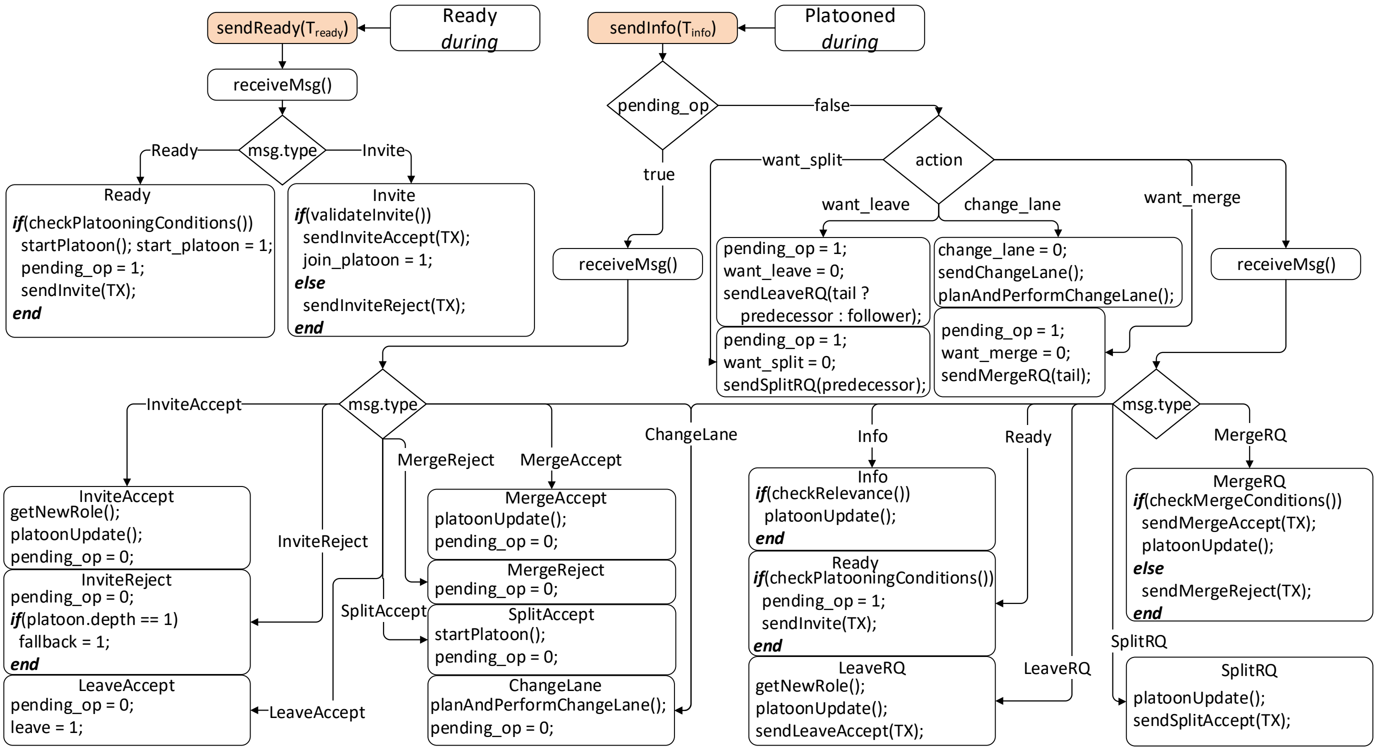


Fig. 2. Platoon management message exchange in states *Ready* and *Platooned*.

their vehicle at the same time. That might be hard to coordinate due to communication delays and retransmissions, especially in long platoons. Furthermore, starting from the head allows to spatially dissolve a platoon: each vehicle may leave the platoon at the same position of a specific road segment.

**Join:** A *Ready* vehicle A is added to an existing platoon. A's *Ready* message is received by vehicle B that is already platooned. It invites A under two conditions: first, no other operation is currently pending, i.e., no other platooning maneuver governed by this vehicle is currently processed. Second, it is B's responsibility to invite A. The responsibility of B depends on its own position within the platoon: in any position, B invites A if A is directly preceding. Additionally, as a tail, B invites A once A is directly following. After the reception of the invitation, A validates the invite (e.g., by checking if the destination is reasonable for its own journey) and accepts or rejects the invite. Once accepted, it changes the state to *Platooned* and A becomes the new predecessor of B. This ensures *join* maneuvers at arbitrary positions: head (joining vehicle gets new leader), mid, or tail.

**Merge:** Contrary to other maneuvers, a *merge* is not processed by vehicles within the same platoon, but by vehicles of different platoons. It is initiated by the leader of the following platoon (A) and is addressed to the tail of the leading platoon (B). After B decides about the reasonability of the merge, it is accepted or rejected. Rejection does not introduce any change in the platoon map while acceptance updates both platoon maps.

**Split:** A *split* may be initiated by any vehicle A in a platoon with at least one predecessor and one follower and is addressed to the predecessor (B). The *split* request can not be rejected. B updates its map and sends an accept to inform A about the successful split: on the receipt of the acceptance message, A starts a new platoon including all its followers.

**Change Lane:** A *change lane* procedure may occur due to external events, mostly caused by road scenarios, e.g., current lane going into different direction then the destination. The procedure is triggered by the leader and can not be rejected. All vehicles of the platoon have to plan and perform the requested lane change.

**Leave:** Removing a vehicle A from the platoon, changes the state of the vehicle to *Not-Platooned*. Reasons for a *leave* may be a driver decision or the end of the common road segment. A *leave* procedure can occur at the head, mid, or tail of the platoon. Vehicle A sends a *leave* request to B, which is the follower or to the predecessor once A is the tail of the platoon. The request can not be rejected. After the reception, B sends an accept and computes its new role in the platoon: if A was the leader or the tail, B becomes the new leader or tail respectively. Else, B receives a new predecessor and updates the platoon map accordingly.

## III. DYNAMIC DISTANCE CONTROL

Vehicles in a platoon try to maintain a predefined distance to the predecessor. The distance is usually the same for all vehicles within a platoon and is selected to satisfy certain requirements. Energy efficiency, for example, favors a lower distance. Vehicles measure the distance to their predecessor and receive further information over V2V communication from their predecessor(s) (e.g., acceleration and velocity). To maintain safety, the inter-vehicle distance may need to increase compared to the predefined distance due to communication delays and measurement uncertainties. Because of driving, vehicles operate in a permanently changing environment (different road scenarios), and external influences may lead to different environmental conditions (e.g., weather, traffic density). Furthermore, vehicles differ by type and thus offer different capabilities (e.g., braking capability) and even same types offer different characteristics (e.g., tire/break conditions, load). *This means every platoon is heterogeneous by nature.* The heterogeneity and the dynamic environment lead to the fact that previously made assumptions about an appropriate reference distance may not be valid anymore: the distance between vehicles has to be adapted dynamically according to the properties described below, named as *Dynamic Platooning*.

**Network properties:** Only a highly reliable network allows close inter-vehicle distances. Once relying on cellular networks, latency, data rate, and coverage differ dependent on the current position and data traffic. Partly, this can be predicted. Low data rate and high latency due to poor coverage is known by radio maps. There, vehicles may plan an increase in distance or prepare a switch to an ad hoc network. In ad hoc networks the reliability mostly depends on the number of transmitting nodes and the load on the network [8]. Knowledge of the current reliability of the underlying network allows updating the reference distance.

**Road:** The current road situation influences the capabilities of a vehicle. This may depend on the current position (uphill/downhill driving) and surface (dry/wet/icy). The current road properties may be sensed directly by the vehicle, may be available through road maps, or could be received by infrastructure (e.g., that there is an oil trace ahead).

**Traffic scenario:** Certain traffic scenarios require an adaptation of the inter-vehicle distance. On highway entries/exits, on rest areas, and in lane merging scenarios, a platoon needs to increase its inter-vehicle distance to allow non platooned vehicles to access/leave the highway, to join the platoon, and to use the lane where the platoon is driving. It may not be necessary to create a gap after every vehicle, e.g., a gap after every 5th vehicle is sufficient. The gap has to be big enough such that a human driver feels comfortable to drive in between the platooned vehicles. Information about specific scenarios may be available due to road maps or may be received from infrastructure.

**Vehicle:** Besides the current state of a vehicle (e.g., position, velocity, acceleration) also static characteristics (e.g., vehicle length) and situation dependent capabilities (e.g., maximum braking due to different load) impact the actuation of the ego vehicle. Own capabilities have to be considered with respect to the capabilities of the predecessor. For example, a longer braking path of the predecessor due to heavy load allows to decrease the distance to the predecessor.

Fig. 3 sums up all external influences that contribute to the determination of the reference distance selected by the ego vehicle. Further it depicts that vehicles keep their view of the environment permanently updated: vehicles' properties and capabilities are exchanged using CAM, while distances are measured by an on-board sensor system. Information about network properties, traffic scenarios, and road conditions are made available by maps, sensed by the vehicle itself, or received over a communication link. Dependent on the used controller, either one or more predecessors may be considered.

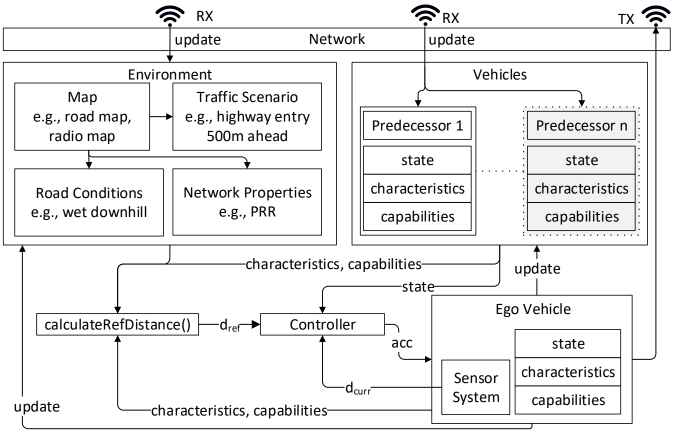


Fig. 3. Dynamic Platooning: Following the reference distance $d_{\text{ref}}$ calculated based on environmental and vehicular constraints.

The desired distance $d_{\text{ref}}$ is a function of combining the mentioned environmental and vehicular impacts (that may interact) and the distances resulting from it, including also a minimum distance $d_{\text{m}}$ we want to guarantee under all circumstances:

$$d_{\text{ref}} = f(d_{\text{network}}, d_{\text{road}}, d_{\text{traffic}}, d_{\text{vehicle}}, d_{\text{m}}) \tag{1}$$

The realization of the determined distance is achieved by smooth trajectory planning of the reference distance using a polynomial: over a time horizon $t_{\text{h}}$ the distance is adapted from the current distance $d_{\text{c}}$ to the target distance $d_{\text{t}}$ generating a new reference distance $d_{\text{ref}}(t)$ for every sampling time step $t$, i.e., one control step.

$$d_{\text{ref}}(t) = d_{\text{c}} + (d_{\text{t}} - d_{\text{c}})[10(\frac{t}{t_{\text{h}}})^3 - 15(\frac{t}{t_{\text{h}}})^4 + 6(\frac{t}{t_{\text{h}}})^5] \tag{2}$$

This enables comfort, even without using precontrol.

## IV. SIMULATION

For simulation purposes the discrete event simulator Omnet++ is used, coupled over the TraCI interface with SUMO (Simulation of Urban Mobility). V2V communication relies on 802.11p that is provided by the framework Artery [16] that builds on top of Veins and Vanetza (see Fig. 4). Using the simulation environment, we test dynamic platooning in two different scenarios: first, we investigate the in Section II presented platoon formation and join maneuver in a highway entry scenario using smooth trajectory planning as presented in Section III. Second, dynamic platooning is applied in an emergency braking scenario relying on a network with varying packet reception rate (PRR).

### *A. Framework*

Artery allows defining several services that work independently of each other. The so called *CaService* implements

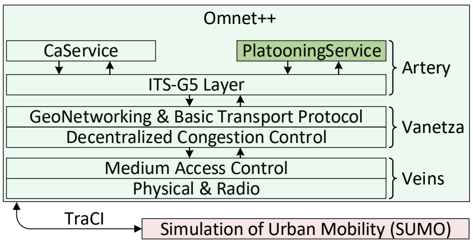


Fig. 4. Simulation Environment in Omnet++ and Sumo (adapted from [16]). PlatooningService as a part of Artery, building on top of Vanetza and Veins.

the current standard for CAM dissemination, specified in EN 302 637-2 V1.3.2. Using those messages, vehicles get aware of each other. We define the new service *PlatooningService* supporting platoon management and dynamic platooning. So far, the service implements following functionalities:

**Platoon formation and join:** Vehicles are able to form platoons and join a platoon at arbitrary positions. To do so, the presented state machine in Fig. 1 is implemented, including the messages *Ready*, *Info*, *Invite*, and *InviteAccept*.

**Control of platoon:** For the sake of simplicity, a PD controller with sampling time $T_{\text{Ctrl}}$, is used to control vehicles within the two selected scenarios. The ego vehicle uses the measured distance $d_{\text{curr}}$ and data from its predecessor ($a_1$, $v_1$) to determine distance and velocity error ($e_{\text{d}}$, $e_{\text{v}}$), and further, its own acceleration $a_{\text{E}}$. For platooning applications safety (i.e., collision avoidance) and efficiency (i.e., low acceleration/deceleration) is crucial. In contrast to that, highly accurate tracking of the inter-vehicle distance is often not possible (due to data aging), and also not necessary once the distance is not safety critical. Therefore, we use a exponential scaling factor for the distance, giving it a higher weight once driving at close distances. Tuning can be achieved by constants $k_{\text{d}}$ and $k_{\text{min}}$.

$$\begin{aligned} e_{\text{d}} &= d_{\text{curr}} - d_{\text{ref}},\ e_{\text{v}} = v_1 - v_{\text{E}} \\ a_{\text{E}} &= \frac{a_1}{T_{\text{Ctrl}}} + e_{\text{d}} \cdot [k_{\text{d}} \cdot (e^{-(d_{\text{curr}}-11.0)} + k_{\text{min}})] + e_{\text{v}} \end{aligned} \tag{3}$$

**Dynamic platooning with a specific traffic scenario:** A highway entry increases the inter-vehicle distance by a certain amount in advance, allowing vehicles to join the highway.

**Dynamic platooning with given PRR:** By definition, CAMs are sent with a variable generation rate between 100 ms and 1000 ms. This rate depends on the channel congestion and the rate of change of the system (i.e., change in heading, position, and velocity since the last CAM). Assuming a given PRR, the distance between vehicles have to be adapted to fulfill certain safety requirements. As an example, we consider the requirements specified in ISO26262 ASIL D: the probability of a failure in time should be below $10^{-8}\,\text{h}^{-1}$. Thus, given a PRR, we have to calculate how many packets $x$ in a row have to be lost such that the probability drops below the ASIL D probability:

$$10^{-8} \geq (1 - \text{PRR})^x \rightarrow x \quad \geq \left\lceil \frac{-8}{log(1 - \text{PRR})} \right\rceil \tag{4}$$

The distance $d_{\text{ref}}$ that the ego vehicle has to maintain depends on the minimum distance $d_{\text{m}}$ (should not be reduced), on the traveled path during packet dropout that is determined by the CAM generation rate $T_{\text{GCam}}$, on the speed of the control loop $T_{\text{Ctrl}}$ (i.e., the reaction time of the system once new information is received), and on the velocity of the ego vehicle ($v_{\text{E}}$). Furthermore, a difference in braking paths ($d_{\text{a}}$) has to be considered: calculated by vehicles' velocities ($v_{\text{E}}, v_1 > 0$) and their capabilities, i.e., maximum braking ($a_{\text{E,M}}, a_{\text{1,M}} < 0$). The number of lost packets $x$ is increased by one, as after all lost packets, one packet has to be received successfully. Again, this takes $T_{\text{GCam}}$.

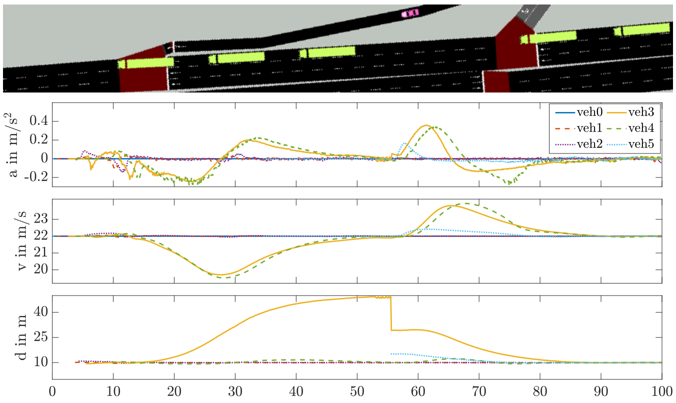


Fig. 5. Highway entry platooning scenario with dynamic distance adaptation ($d$) and resulting velocity ($v$) and acceleration ($a$).

$$\begin{aligned} d_{\text{a}} &= \frac{v_1^2}{2a_{\text{1,M}}} - \frac{v_{\text{E}}^2}{2a_{\text{E,M}}} \\ d_{\text{ref}} &= d_{\text{m}} + max([(x+1)T_{\text{GCam}} + T_{\text{Ctrl}}]v_{\text{E}} + d_{\text{a}}, 0) \end{aligned} \tag{5}$$

### B. Scenarios and Results

Two specific scenarios have been evaluated. In both scenarios, resistances, e.g., from tires, air, or drive train, were neglected. The constants $k_{\text{d}}$ and $k_{\text{min}}$ of the PD controller (3) were set to 0.1.

**Scenario 1:** Five trucks (*veh0*, *veh1*, *veh2*, *veh3*, *veh4*) are driving on a highway, create a platoon and maintain a distance of 10 m (see Figure 5). The distance is measured by a radar sensor. Due to map information, they are aware of an approaching highway entry. At a distance of 900 m before the highway entry (t=11 s), due to the length of the platoon, the decision is made to create a gap of 50 m after the third vehicle: using (2) *veh3* plans an increase of the reference distance over a time horizon of 20 s. A passenger vehicle (*veh5*) is approaching on the highway entry. The driver joins the highway at the created gap (t=53 s). Due to occlusion, *veh3* can not sense the distance to its platoon predecessor (*veh2*) directly but needs to rely on CAM messages. Slow rate of change causes low $T_{\text{GCam}}$ resulting in jittering distance. *Veh5* then decides to platoon: it sends a *Ready* message that triggers a decentralized join procedure as presented in Fig. 2. *Veh5* accepts the invite of *veh3* and becomes its new predecessor. Both vehicles adapt their distance to 10 m, again using (2).

**Scenario 2:** Two trucks are driving platooned on a highway with a velocity of $22\,\text{m}\,\text{s}^{-1}$. The leading truck is on the way to pick up goods and is able to brake with a maximum of

$-7\,\mathrm{m\,s^{-2}}$. The following truck carries heavy load, allowing maximum braking of $-5\,\mathrm{m\,s^{-2}}$. As a minimum distance $d_\mathrm{m}$, $5\,\mathrm{m}$ is defined. After $15\,\mathrm{s}$, the leader detects a vehicle on the emergency track and its driver on the leader's lane. To avoid an accident, the leader utilizes its maximum braking capability. To prevent a rear-end collision, also the follower has to brake. In a emergency braking scenario, a high rate of change is achieved leading to a CAM generation rate of $100\,\mathrm{ms}$. We investigate the described scenario using a control speed of $0.1\,\mathrm{s}$, starting with a PRR of 1, i.e., $100\,\%$, and decrease the PRR rate in each simulation round.

To test if the calculated distance corresponds to ASIL D, we drop $x$ CAM packets once the emergency brake occurs, as computed in (4). In case of a PRR of 1, $x$ is set to 0. Table I shows calculated distance $d_\mathrm{calc}$ (5), distance $d_\mathrm{brake}$ where the braking actually started, and the distance between vehicles $d_\mathrm{stop}$ once both stopped. The results show that even with an unreliable network, due to dynamic distance adaption, a rear-end collision could be avoided fulfilling ASIL D requirements. Furthermore, $d_\mathrm{stop}$ is still larger than the specified minimum distance $d_\mathrm{m}$. Divergences from $d_\mathrm{m}$ are due to worst case assumptions: Equation (5) assumes that the first lost packet would arrive $T_\mathrm{GCam}$ after the emergency brake occurred. Moreover, it assumes that the controller is executed $T_\mathrm{Ctrl}$ after the successful reception of a CAM. As both values are $0.1\,\mathrm{s}$, deviations up to $0.2 \cdot v_\mathrm{E}$ are possible. By lowering the PRR, we observe an increased stopping distance: during the time of $x$ packet losses in a row, no update about the predecessors acceleration and velocity is received. Still, the radar sensor measures a decreasing distance, introducing an increasing error between reference distance $d_\mathrm{ref}$ and current distance $d_\mathrm{curr}$. A higher error decreases the acceleration due to the used PD controller (3).

TABLE I
SCENARIO 2: DYNAMIC DISTANCE WITH PRR AND STOPPING DISTANCE.

| PRR in % | $d_\mathrm{calc}$ in m | $d_\mathrm{brake}$ in m | $d_\mathrm{stop}$ in m |
|---|---|---|---|
| 100 | 23.2286 | 23.2775 | 7.4269 |
| 90 | 40.8286 | 40.8318 | 7.6138 |
| 80 | 49.6286 | 49.3340 | 7.9975 |
| 70 | 58.4286 | 58.0276 | 8.7452 |

## V. CONCLUSION

A decentralized architecture is proposed allowing to create, maintain, and dissolve heterogeneous platoons including necessary platoon maneuvers and the corresponding message flow. Inter-vehicle distances are determined individually by each vehicle accounting for changing environment and vehicles' properties. The architecture is evaluated in a highway entry and an emergency brake scenario implementing our own service on top of Artery in Omnet++: it shows that a dynamic and individual adaptation of the inter-vehicle distance allows targeting specific traffic scenarios and guarantees safety even in lossy communication networks. The realization of the dynamic distance is performed using smooth trajectory planning.

Further work remains in the implementation and evaluation of the proposed maneuvers (*leave*, *dissolve*, *merge*, *split*, and *lane change*) and the caused platoon map update. Beyond that, the influence of road conditions and further traffic scenarios need to be investigated. Furthermore, research will be performed in the determination of the current network properties dealing with both ad hoc and cellular networks. Considering the controller, different approaches will be compared.

## ACKNOWLEDGMENT

The authors kindly acknowledge financial support of the COMET K2 – Competence Centers for Excellent Technologies Programme of the Federal Ministry for Transport, Innovation and Technology (bmvit), the Federal Ministry for Digital, Business and Enterprise (bmdw), the Austrian Research Promotion Agency (FFG), the Province of Styria and the Styrian Business Promotion Agency (SFG), and express their thanks to their partners, AVL List GmbH and to Infineon Technologies Austria AG for financial support.

## REFERENCES

[1] C. Bonnet and H. Fritz, "Fuel consumption reduction in a platoon: Experimental results with two electronically coupled trucks at close spacing," SAE Technical Paper, Tech. Rep., 2000.

[2] A. Rupp and M. Stolz, "Survey on control schemes for automated driving on highways," in *Automated driving*. Springer, 2017, pp. 43–69.

[3] G. Guo and W. Yue, "Hierarchical platoon control with heterogeneous information feedback," *IET control theory & applications*, vol. 5, no. 15, pp. 1766–1781, 2011.

[4] A. Alam, B. Besselink, V. Turri, J. Martensson, and K. H. Johansson, "Heavy-duty vehicle platooning for sustainable freight transportation: A cooperative method to enhance safety and efficiency," *IEEE Control Systems Magazine*, vol. 35, no. 6, pp. 34–56, 2015.

[5] V. Turri, B. Besselink, and K. H. Johansson, "Cooperative look-ahead control for fuel-efficient and safe heavy-duty vehicle platooning," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 1, pp. 12–28, 2017.

[6] K.-Y. Liang, S. van de Hoef, H. Terelius, V. Turri, B. Besselink, J. Mårtensson, and K. H. Johansson, "Networked control challenges in collaborative road freight transport," *European Journal of Control*, vol. 30, pp. 2–14, 2016.

[7] V. Vukadinovic, K. Bakowski, P. Marsch, I. D. Garcia, H. Xu, M. Sybis, P. Sroka, K. Wesolowski, D. Lister, and I. Thibault, "3gpp c-v2x and ieee 802.11p for vehicle-to-vehicle communications in highway platooning scenarios," *Ad Hoc Networks*, vol. 74, pp. 17–29, 2018.

[8] Z. H. Mir and F. Filali, "Lte and ieee 802.11 p for vehicular networking: a performance evaluation," *EURASIP Journal on Wireless Communications and Networking*, vol. 2014, no. 1, p. 89, 2014.

[9] Y. Qi and T. Mach, "Qos and coverage aware dynamic high density vehicle platooning (hdvp)." [Online]. Available: http://arxiv.org/pdf/1807.07337v1

[10] G. Nardini, A. Virdis, C. Campolo, A. Molinaro, and G. Stea, "Cellular-v2x communications for platooning: Design and evaluation," *Sensors (Basel, Switzerland)*, vol. 18, no. 5, 2018.

[11] T. Lorenzen and H. Tchouankem, "Evaluation of an awareness control algorithm for vanets based on etsi en 302 637-2 v1.3.2," in *2015 IEEE International Conference on Communication Workshop (ICCW)*. IEEE, 08.06.2015 - 12.06.2015, pp. 2458–2464.

[12] M. Segata, B. Bloessl, S. Joerer, F. Dressler, and R. Lo Cigno, "Supporting platooning maneuvers through ivc: An initial protocol analysis for the join maneuver," in *2014 11th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*. IEEE, 02.04.2014 - 04.04.2014, pp. 130–137.

[13] C. Bergenhem, Q. Huang, A. Benmimoun, and T. Robinson, "Challenges of platooning on public motorways," in *17th world congress on intelligent transport systems*, 2010, pp. 1–12.

[14] M. Amoozadeh, H. Deng, C.-N. Chuah, H. M. Zhang, and D. Ghosal, "Platoon management with cooperative adaptive cruise control enabled by vanet," *Vehicular Communications*, vol. 2, no. 2, pp. 110–123, 2015.

[15] A. Alam, J. Mårtensson, and K. H. Johansson, "Experimental evaluation of decentralized cooperative cruise control for heavy-duty vehicle platooning," *Control Engineering Practice*, vol. 38, pp. 11–25, 2015.

[16] R. Riebl, H.-J. Gunther, C. Facchi, and L. Wolf, "Artery: Extending veins for vanet applications," in *2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*. IEEE, 03.06.2015 - 05.06.2015, pp. 450–456.