# Platoon Management with Cooperative Adaptive Cruise Control Enabled by VANET

Mani Amoozadeh[1], Hui Deng[2], Chen-Nee Chuah[1], H. Michael Zhang[2], Dipak Ghosal[3]
[1]Department of Electrical and Computer Engineering, University of California, Davis, CA, USA
[2]Department of Civil and Environmental Engineering, University of California, Davis, CA, USA
[3]Department of Computer Science, University of California, Davis, CA, USA
{maniam,huideng,chuah,hmzhang,dghosal}@ucdavis.edu

## Abstract

Previous studies have shown the ability of vehicle platooning to improve highway safety and throughput. With Vehicular Ad-hoc Network (VANET) and Cooperative Adaptive Cruise Control (CACC) system, vehicle platooning with small headway becomes feasible. In this paper, we developed a platoon management protocol for CACC vehicles based on wireless communication through VANET. This protocol includes three basic platooning maneuvers and a set of *micro-commands* to accomplish these maneuvers. Various platooning operations such as vehicle entry and vehicle (including platoon leader) leaving can be captured by these basic platoon maneuvers. The protocol operation is described in detail using various *Finite State Machines (FSM)*, and can be applied in collaborative driving and intelligent highway systems. This protocol is implemented in an integrated simulation platform, VENTOS, which is developed based on SUMO and OMNET++. The validity and effectiveness of our approach is shown by means of simulations, and different platooning setting are calibrated.

*Keywords:* VANET, Cooperative Adaptive Cruise Control (CACC), Automatic Cruise Control (ACC), Platoon Management, Platoon maneuver

## 1. Introduction

Vehicle platooning is a technique where highway traffic is organized into groups of close-following vehicles called *platoon* or *convoy*. The most widely studied platoon configuration in transportation is the column, also known as *road trains*, although other types of formations can also be considered [1]. Platooning enables vehicles to drive closer (maintain a smaller headway) than normal vehicles with the same speed, which improves traffic throughput as well as homogeneity. Additionally, safety is enhanced due to small speed variation and relative low impact velocity in collisions. Platooning is not only a promising way to improve traffic efficiency and safety, but can also reduce fuel consumption and emissions due to air drag reduction [2]. Platooning can be regarded as an eco-driving strategy and is most effective for Heavy-duty Vehicles (HDVs). It is considered as a promising solution to reduce fuel consumption in HDVs [3, 4, 5], and is getting increasing attention from private fleets and commercial carriers [6]. Last but not least, platooning facilitates more efficient information dissemination and sharing among vehicles in the same platoon [7].

While platooning was originally designed for Automated Highway System (AHS), the improvements in wireless communication and vehicle control technology make platooning feasible for partial automated vehicles, such as *Cooperative Adaptive Cruise Control (CACC)* vehicles. The ability of CACC vehicles to follow closely enable them to drive in tight platoons, and consequently increase highway throughput. While the technical feasibility of platooning has been analyzed worldwide under various projects, the details of the platooning vary among projects since there are different goals and motivations for doing platooning [8].

Although platooning offers a lot of benefits, it requires cooperation between vehicles with the help of a platoon management protocol. A well-developed platoon management protocol is important to ensure good CACC platooning performance and should be verified before using in real-world applications. Our main contributions in this paper are:

**Developing a platoon management protocol:** We have developed a platoon management protocol based on Vehicular Ad-hoc Network (VANET) and CACC vehicles that uses three basic platoon maneuvers: merge, split and lane-change. These three basic maneuvers can be used to accomplish various platoon operations, such as vehicle entry, platoon leader leave, and follower leave. Our protocol is based on Vehicle-to-Vehicle (V2V) communication with single-hop beacon messages as well as event-driven messages to coordinate the maneuvers with other neighboring vehicles.

**Implementation of CACC platooning protocol:** VENTOS (VEhicular NeTwork Open Simulator) [9] is an integrated open-source simulator that we have developed to study VANET-based traffic applications. CACC car-following model based on one-vehicle look-ahead communication and platoon management protocol are implemented in VENTOS in order to study CACC platooning performance. We consider lane change as a manual driving behavior and the default lane change model, LC2013 [10], in SUMO is adopted as the lateral control logic.

**Verification of our CACC platooning protocol:** Using VENTOS we have tested our protocol for different platooning scenarios. We also performed sensitivity analysis of our CACC platooning settings, including platoon communication structure, inter-platoon spacing, intra-platoon CACC time-gap setting, and platoon size.

## 2. Related Works

The idea of organizing traffic in platoons to dramatically increase capacity is originally proposed in [11] by PATH for *Intelligent Vehicle Highway System (IVHS)*, and was successfully demonstrated by National Automated Highway Systems Consortium (NAHSC) using eight cars on I-15 in San Diego, CA, in 1997 [12]. They propose a system architecture where control tasks are arranged in a five-layer hierarchy. Physical, regulation, and coordination layers are distributed among controllers on each vehicle, whereas link and network layer control groups of vehicles. Our proposed platoon management protocol resides in the coordination layer[1], and interacts with link and regulation layers.

Over the years, other architectures related to collaborative driving have been proposed. The architecture in [13] views collaborative driving as a group of dolphins that swim without collision while communicating with each other, and consists of control layer, management layer, and traffic control layer. The potential of this architecture in cooperative driving was demonstrated using five automated vehicle platooning in November, 2000 on a oval-shaped 3.2 km-long test track as part of the DEMO2000 project [14]. The architecture in [15] proposed a hierarchical architecture with guidance layer, management layer, and traffic control layer. Our platoon management protocol fits nicely into the management layer of the aforementioned architectures.

Fernandes et al. [16] present a simulation engine for platoons of autonomous vehicles in SUMO by implementing a longitudinal control in vehicles. They use a relatively simple controller that considers only the gap control mode. On top of that, no actual platoon management protocol has been implemented. Thus there is no coordination between vehicles in maneuvers.

Michaud et al. [17] discuss different coordination strategies of automated vehicles in platoons that is mostly focused on communication patterns between vehicles in centralized or decentralized[2] fashion. They use a mobile robotic platform to emulate platooning conditions that does not model real vehicle dynamics. Some important aspects such as ensuring string stability is not considered in the design. Another example of decentralized platoon coordination is presented in Auto21 project [18, 19] by using an agent teamwork model based on a multiagent architecture, known as STEAM.

Segata et al. [20] develop an integrated simulator for studying strategies and protocols in platooning scenarios. To the best of our knowledge, this is the first attempt in designing a high-level platoon management protocol leveraging wireless V2V communication with IEEE 802.11p in VANET-enabled vehicles. An extension to this work [21] focuses on the join maneuver only and analyzes the interferences caused by non-automated vehicles as well as analyzing the impact of packet loss on the failure rate of the maneuver. However, the topic has not been touched upon with great details, and more simulation scenarios are needed.

Our work presents a comprehensive longitudinal control system for CACC vehicles to better model the real vehicle dynamics that is neglected in [17]. Vehicles communicate wirelessly using IEEE 802.11p, and platooning is done through a platoon management protocol that coordinates all maneuvers that is not discussed in [16]. In addition to the join maneuver discussed in [21], our designed protocol permits split, leader leave, and follower leave maneuvers with the help of special messages called *micro-commands*. Moreover, under communication failure, the protocol is able to re-transmit messages using different timers incorporated in the algorithm. The validity of our approach is shown by means of packet-level simulations described in Section 6. Through a detailed simulation study we show the effectiveness of our designed protocol and study the platooning behavior, traffic flow throughput, duration of each platooning maneuvers, and the impact of communication failure.

## 3. Platoon Management Framework

A platoon is composed of a *platoon leader* which is normally the first vehicle in the platoon and one or more *followers* that pursue each other closely. In many experimental studies, due to safety reasons, platoon leader is driven by a trained professional driver. Following vehicles are driven fully automatically, allowing the drivers to

[1] Coordination layer has three tasks: (1) to determine which maneuvers the vehicle should execute, so that its trajectory is close to its assigned path from the link layer; (2) coordinate that maneuver with the coordination layers of neighboring vehicles to ensure safety; and (3) to supervise its regulation layer in the execution of a trajectory corresponding to the maneuver [11].

[2] In decentralized coordination, platooning decisions are distributed locally among platoon members. Platoon members can react autonomously, and directly communicate with each other.

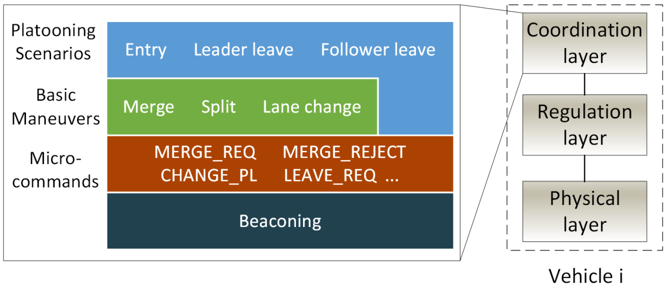


Figure 1: Platoon management protocol resides in the coordination layer of each vehicle and uses wireless V2V communication to exchange micro-commands between neighboring vehicles in order to coordinate different maneuvers.

perform tasks other than driving such as using a mobile phone [22]. Our designed platoon management protocol shown in Fig. 1 resides in the coordination layer of each platoon member and is responsible for coordinating different maneuvers with neighboring vehicles.

In our study, vehicles in the platoon are CACC-capable and participate in information-sharing and cooperative driving through VANET communications. CACC system is basically an enhancement of Adaptive Cruise Control (ACC)[3] which incorporates wireless V2V communication to access rich preview information about the surrounding vehicles [23]. This leads to tighter following gaps and faster response to changes than ACC. The CACC control logic which falls in the regulation and physical layer of each vehicle is also shown in Fig. 1 and is described in Section 3.3.

### *3.1. Platoon Management Protocol*

The platoon management protocol supports three basic or elementary maneuvers: *merge*, *split*, and *lane change*. In merge, two platoons, traveling on the same lane, merge to form one big platoon. In split, one platoon (with at least two vehicles) separates at a specific position to form two smaller platoons. Finally, lane change permits a one-vehicle platoon (free agent) to change lane. We can deal with more complex platooning scenarios using the aforementioned basic maneuvers. For instance, *platoon follower leave* can be performed using a sequence of split, lane change, and merge.

Due to the distributed control design, each maneuver is coordinated by exchanging a sequence of *micro-command* messages between platoon management protocols in different vehicles. In total, we have defined a set of seventeen micro-commands that are detailed in Appendix A. Wireless Vehicle-to-Vehicle (V2V) communication is achieved using *Dedicated Short Range Communication (DSRC)* which has been standardized in IEEE 802.11p [24].

DSRC operating at 5.9 GHz is designed to support a large variety of applications. It supports high data rates, and has a transmission range of 100 to 1000 meters which makes it suitable for information dissemination and sharing among vehicles in a platoon. Although DSRC has been the prominent Inter-vehicle Communication (IVC) technology in ITS context, the performance of DSRC broadcasting may raise some reliability concerns [25].

The platoon management protocol uses a *centralized* platoon coordination approach where all communications are coordinated by the platoon leader. Followers take orders and send requests from/to the platoon leader. We believe that centralized platoon coordination is fast, scalable, and does not make platoon leader a bottleneck since the frequency at which platooning decision is exchanged is low (once every minute [11]). Moreover, only the platoon leader stores and manages the platoon configuration. This enhances privacy in situations where followers should not have access to the platoon configuration, since they dynamically enter and exit the platoon.

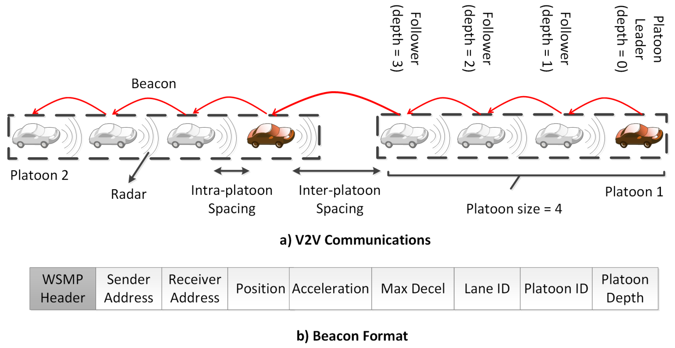
Consider an example in which vehicle A relinquishes control, and joins a platoon. The platoon begins its journey on the highway and performs necessary maneuvers. Eventually, as vehicle A approaches its destination, it leaves the platoon. In the centralized platoon coordination, all the necessary platoon configuration data (Table A.2) are stored in the platoon leader and are kept hidden from the followers. Vehicles A does not need to know the platoon size, type and destination of other vehicles, etc. Sharing of platoon configuration data is only done when the platoon leader leaves in which case the old leader passes all the necessary data to the new leader.

### *3.2. CACC Vehicle Control for Platooning*

In our study, CACC vehicles use a simple one-vehicle look-ahead communication scheme as illustrated in Fig. 2a where each vehicle is listening to beacon messages sent wirelessly from its immediate preceding vehicle. The vehicles then utilize the speed, position, acceleration and other information embedded in these beacon messages to achieve a distributed longitudinal control. Platoon leader is also assumed to be a CACC vehicle and listens to its preceding vehicle (if present) in the front platoon. The longitudinal control logic consists of multiple operation modes, and the system transits between these modes in order to generate the desired acceleration. We provide more details in Section 3.3.

Parameter exchange in CACC vehicles is done through *beacons*. Beacons are periodic single-hop messages broadcasted by each vehicle in VANET, and are used for cooperative applications. The beacon message format is shown in Fig. 2b. The size of the beacon message (including

[3] ACC vehicles allow drivers to maintain a desired following gap with respect to a preceding vehicle based on range sensor (radar or LIDAR) measurements of distance and speed difference in order to achieve longitudinal control by actuating the brake and throttle of the vehicle in an automated manner. These automated responses can occur faster than human reactions to road events. Hence, this technology allows equipped vehicles to safely travel closer together increasing the road capacity and improving the fuel efficiency of the vehicles.



a) V2V Communications

| WSMP Header | Sender Address | Receiver Address | Position | Acceleration | Max Decel | Lane ID | Platoon ID | Platoon Depth |
|---|---|---|---|---|---|---|---|---|

b) Beacon Format

Figure 2: One-vehicle look-ahead communication in CACC platoon. $i$th vehicle receives information embedded in beacon messages from $(i-1)$th vehicle using V2V wireless communication and feeds it into the longitudinal control system (detailed in Section 3.3) to maintaining a safe gap from the preceding vehicle. All vehicles are equipped with radar to measure the space-gap and velocity of the preceding vehicle.

header) is fixed and equal to 96 Bytes. Beaconing interval is 0.1s, and the first beacon broadcast in a vehicle is done with a random offset. This leads to dramatic reduction in number of collisions in the network, especially in dense vehicular traffic scenarios.

To support platooning, *Platoon Id* and *Platoon depth* fields are added to the beacon messages. Platoon Id is a unique string that is used to differentiate between different platoons, and platoon depth is an integer that shows the vehicle position within the platoon. Platoon leader has depth of 0, and it increases as we go farther. Similar to previous studies [26], we set platoon Id to be vehicle Id of the platoon leader. Rather than using a fixed platoon Id, we can generate a *platoon-filter* [27] which is based on Bloom filter which gives a space-efficient probabilistic data structure for testing if a vehicle is a member of a platoon. Platoon leader is responsible for updating the platoon-filter when follower(s) join or leave. Platoon-filter can be also used as an intra-platoon multicast address, and can assist in intra-platoon packet-loss detection and cooperative re-transmission [28].

### *3.3. CACC Longitudinal Control Logic*

A CACC longitudinal control system is normally designed to have an upper-level controller and lower-level controller which correspond to regulation and physical layer in Fig. 1 respectively. The lower-layer controller determines the throttle and/or brake commands required to track the desired acceleration [29]. Our upper-level controller has multiple operation modes (*Speed Control mode (SC)*, *Gap Control mode (GC)* and *Collision Avoidance mode (CA)*), and the system transits between these modes in order to generate the desired acceleration. The different parameters of the longitudinal control system are shown in Table 1.

Table 1: Default parameters for CACC vehicle platooning

| **Parameter** | **Description** | **Value** |
|---|---|---|
| $L_v$ | Vehicle length | 5.0 m |
| $G_{min}$ | Min space-gap | 2.0 m |
| $T_g$ | Intra-platoon time-gap | 0.55 s |
| $T_p$ | Inter-platoon time-gap | 3.5 s |
| $\tau$ | Reaction time/controller delay | 0.4 s |
| $V_{max}$ | Max speed | 30 m/s |
| $V_{int}$ | Intended speed | 20 m/s |
| $A^{max}$ | Maximum acceleration | 3 m/s$^2$ |
| $D^{max}$ | Maximum deceleration | 5 m/s$^2$ |
| $A^{cf}$ | comfort acceleration limit | 2 m/s$^2$ |
| $D^{cf}$ | comfort deceleration limit | 3 m/s$^2$ |
| $K_{sc}$ | Speed control gain | 0.4 $s^{-1}$ |
| $K_a$ | Acceleration gain | 0.66 $s^{-1}$ |
| $K_v$ | Speed gain | 0.99 $s^{-1}$ |
| $K_g$ | Gap gain | 4.08 $s^{-2}$ |
| $C_w$ | Communication frequency | 10 Hz |

Each vehicle in the platoon tries to maintain a safe space-gap with its preceding vehicle. Safe space-gap denoted by $g_{safe}$, is determined by speed and maximum deceleration ability of the vehicle and its preceding vehicle, and is determined by

$$g_{safe} = v \times 0.1 + \frac{v^2}{2D^{max}} - \frac{v_p^2}{2D_p^{max}} + 1.0 \qquad (1)$$

where 1.0 is the minimum space-gap. As soon as the instantaneous space-gap $g \leq g_{safe}$, vehicle switches to CA mode and uses maximum deceleration $D^{max}$ to avoid collision. As long as the instantaneous space-gap $g > g_{safe}$, vehicle stays in either SC mode or GC mode. The SC mode captures the *free driving* behavior in which the vehicle tries to maintain the intended speed $V_{int}$. The desired acceleration for SC mode is given by,

$$a_v^* = K_{sc}(V_{int} - v) \qquad (2)$$

where $v$ is the following vehicle speed, and $K_{sc}$ is the speed control gain. The GC mode captures the *following driving* behavior, and the vehicle tries to follow its preceding vehicle with time-gap $T_g$. The desired acceleration in GC mode is given by,

$$a_g^* = K_a a_p + K_v\,(v_p - v) + K_g\,(g - G_{min} - vT_g) \qquad (3)$$

where $a_p$ is the preceding vehicle's acceleration, $v$ is the following vehicle speed, $v_p$ is the preceding vehicle speed, $g$ is the space-gap, $T_g$ is the desired time-gap, $G_{min}$ is the minimum standstill gap, and $K_g$, $K_v$ and $K_a$ are positive gains for gap, speed and acceleration, respectively. The control parameter $a_p$ is obtained using wireless communication, while $g$, and $v_p$ are obtained using radar measurement. While obtaining space-gap and velocity data using

V2V communication might be faster and more accurate, packet loss can be a major problem that prevents obtaining timely data. This can potentially leads to rear-end collision since stale data does not allow the CA mode to intervene.

In homogeneous vehicles case, $G_{min}$ is set to be 2.0 m. When vehicles are non-homogeneous, and the following vehicle has lower deceleration capability than the preceding vehicle, $G_{min}$ is increased and is given by,

$$G_{min} = 2.0 + \max(0,\ \frac{V_{int}^2}{2D^{max}} - \frac{V_{int}^2}{2D_p^{max}}) \tag{4}$$

where $D^{max}$ and $D_p^{max}$ are maximum deceleration of current and preceding vehicle respectively. The final desired acceleration $a_{des}$ is the minimum of $a_v^*$ and $a_g^*$ and is given by

$$a_{des} = min(a_v^*, a_g^*) \tag{5}$$

The desired acceleration $a_{des}$ computed above cannot be applied immediately. There will be a certain actuation time lag $\tau$ which is due to the dynamics of the vehicle. For simplicity, we modeled this lag as a first order time lag. Thus the actual acceleration $a_{control}$ is given by,

$$a_{control} = \frac{a_{des} - a}{\tau} \times \Delta t + a \tag{6}$$

where $\Delta t$ is the simulation time step, and is equal to 0.1 s. For driver comfort, $a_{control}$ is bounded by $A^{cf}$ and $D^{cf}$. New acceleration $a^+$, and new follow speed $v^+$ are given by,

$$a^+ = mid(A^{cf}, a_{control}, D^{cf}) \tag{7a}$$

$$v^+ = max(0, v + a^+ \times \Delta t) \tag{7b}$$

### *3.4. Platooning Setting*

As illustrated in Fig. 2, there are three critical parameter settings for CACC platooning: *intra-platoon spacing*, *inter-platoon spacing*, and *platoon size*.

- **Intra-platoon spacing:** Inside a platoon all the vehicles follow the leader with a small intra-platoon separation which corresponds to CACC time-gap setting. Vehicle time-gap setting is limited by *string stability*. String stability, also referred to as platoon stability and asymptotic stability, concerns with the propagation of disturbance in a string of vehicles. String stable means disturbance damps out when propagating to upstream vehicles. Small time-gap setting contributes to high throughput but may compromise the robustness of CACC platoon.

- **Inter-platoon spacing:** Inter-platoon spacing is the separation between two different platoons which is assumed to be large. Large inter-platoon spacing gap is assigned to accommodate lane changing and suppress traffic disturbances for platooning to minimize the collision between platoons.

- **Platoon size:** Platoon size shows the number of vehicles in the platoon, and it dynamically changes as vehicles enter and exit the platoon. Although larger platoon size can increase throughput, it may negatively affect platoon flexibility and increase the requirement of *traffic flow stability.* Traffic flow stability concerns with disturbance propagation across platoons. Traffic flow is unstable if instability in one platoon transfer to the next platoon and disturbances grow in amplitude. In the initial design of fully automated highway systems [11], platoon size can be as large as 20. In SARTRE project [22], the maximum recommended platoon size is 15.

## 4. Basic Platoon Maneuvers

Each platooning maneuver is coordinated by a structured exchange of messages among relevant neighboring vehicles or Road-Side Units (RSUs). These message exchanges can be defined precisely using a *Finite State Machine (FSM)* that progresses through a series of *states* in response to various *events.* In this section, we will explain how the platoon management protocol performs the three basic maneuvers namely, merge, split, and lane change.

A special lane is reserved for platooned vehicles in a highway and each platoon can perform different maneuvers to maintain the optimal platoon size that is usually dictated by the RSU. This is illustrated in Fig. 3 where the leftmost lane (lane 1) is reserved for platooned vehicles. A platoon of 4 cars (platoon A) is traveling on lane 1 with $C_1^A$ acting as the leader. Vehicles on other lanes are driving under manual driving or ACC system. A platoon-enabled vehicle entering the highway can change to the special lane and begins its journey as a platoon member, and may perform different maneuvers before reaching its destination.

It is important to differentiate between *platoon-enabled*, *non-platooned*, and *platooned* vehicles. A platoon-enabled vehicle has all the required hardware and software, and can be either a non-platooned or platooned vehicle. A non-platooned vehicle is not part of any platoons, and it is traveling under manual driving or ACC system, whereas a platooned vehicle is a member of a platoon, and can be either a leader or follower. Any platoon-enabled vehicle traveling on lane 1 which has not joined any platoons yet is a *free agent.* A free agent can be considered as a one-vehicle platoon with platoon size of 1.

Maneuvers can happen at any point along the highway, but we only allow one maneuver at a time. Permitting more than one maneuver simultaneously makes the coordination task more complex, and leads to a larger FSM.

### *4.1. Merge Maneuver*

In merge maneuver, two platoons, traveling on the same lane, merge to form one platoon. This maneuver is always initiated by the platoon leader of the rear platoon

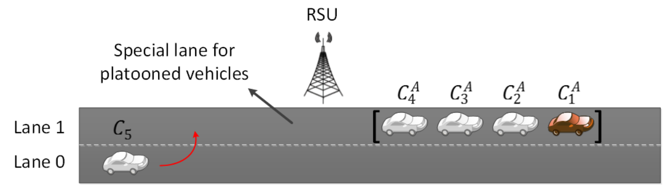

RSU
Special lane for platooned vehicles
Lane 1
Lane 0
$C_5$
$C_4^A$ $C_3^A$ $C_2^A$ $C_1^A$

Figure 3: Two-lane highway with leftmost lane reserved for platooned vehicles. Vehicle $C_5$ is about to perform an entry maneuver to lane 1

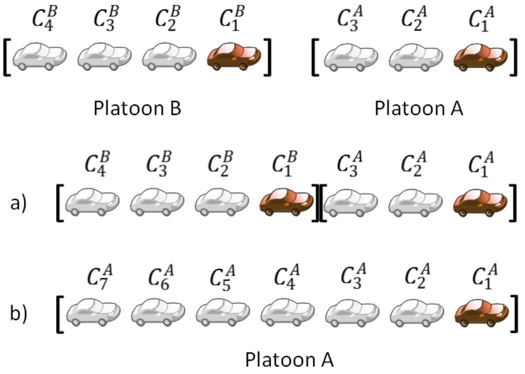

$C_4^B$ $C_3^B$ $C_2^B$ $C_1^B$
$C_3^A$ $C_2^A$ $C_1^A$
Platoon B
Platoon A
a)
$C_4^B$ $C_3^B$ $C_2^B$ $C_1^B$ $C_3^A$ $C_2^A$ $C_1^A$
b)
$C_7^A$ $C_6^A$ $C_5^A$ $C_4^A$ $C_3^A$ $C_2^A$ $C_1^A$
Platoon A

Figure 4: Platoon B is merged into platoon A, and the final platoon size is 7 (leaders are shown in red)

when the platoon size is less than the optimal platoon size. Assume we have two platoons, A and B as shown in Fig. 4. Platoon B is the rear platoon with platoon leader $C_1^B$ and consists of 4 vehicles, whereas platoon A is the front platoon with platoon leader $C_1^A$ and consists of 3 vehicles. Also assume that the optimal platoon size is 8. The following steps show how the merge maneuver is performed:

**1. Merge request:** $C_1^B$ receives a beacon message from one of the platoon members in the front platoon, and can initiate a merge maneuver, since $C_1^B$'s platoon size is less than the optimal platoon size. It extracts platoon Id of the front platoon (which is the vehicle Id of $C_1^A$) from the beacon and sends a unicast MERGE_REQ to the front platoon leader ($C_1^A$).

**2. Merge response:** The front platoon leader, $C_1^A$, can accept or reject the merge request. If $C_1^A$ is busy performing a maneuver, then it sends MERGE_REJECT to either reject or delay the merging maneuver. The latter case is called a *weak reject*, and $C_1^B$ has this option to send the MERGE_REQ again later. $C_1^A$ accepts the merge request if the final platoon size does not exceed the optimal platoon size.

**3. Merge execution:** Upon MERGE_ACCEPT reception, $C_1^B$ reduces its time-gap to intra-platoon spacing, and catches up with the front platoon (Fig. 4a). $C_1^B$ sends CHANGE_PL to all its followers to change the platoon leader to $C_1^A$ (Fig. 4b). Now the followers start listening to $C_1^A$. Finally, $C_1^B$ sends a MERGE_DONE to $C_1^A$, and changes its state from leader to follower.

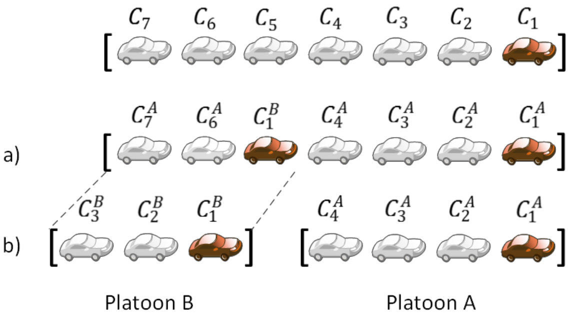

$C_7$ $C_6$ $C_5$ $C_4$ $C_3$ $C_2$ $C_1$
a)
$C_7^A$ $C_6^A$ $C_1^B$ $C_4^A$ $C_3^A$ $C_2^A$ $C_1^A$
b)
$C_3^B$ $C_2^B$ $C_1^B$ $C_4^A$ $C_3^A$ $C_2^A$ $C_1^A$
Platoon B
Platoon A

Figure 6: Platoon of size 7 is split into platoon A with size 4 and platoon B with size 3 (leaders are shown in red)

The state machine for merge maneuver is shown in Fig. 5. States with a small black triangle at the corner are transient states where the state machine spends zero time. Both platoon leaders in the rear and front platoon follow this FSM, but they take a different path through the states. In other words, the operation of the protocol is not symmetric.

### *4.2. Platoon Split Maneuver*

In split maneuver, one platoon (with at least two vehicles) separates at a specific position to form two smaller platoons. Similar to merge maneuver, platoon split maneuver is always initiated by the platoon leader. When platoon size exceeds the optimal platoon size, split maneuver can be used to break the platoon into two smaller parts. Assume we have a platoon of size 7, and we want to split it into platoon A with size 4 and platoon B with size 3, as shown in Fig. 6. The following steps show how the platoon split maneuver is performed:

**1. Split request:** The platoon leader, $C_1$, initiates the split maneuver by sending a SPLIT_REQ message to the splitting vehicle ($C_5$ in the above example).

**2. Split response:** $C_5$ either accepts or rejects the split request. In the later case, $C_5$, can include the reason for rejecting the split maneuver into the SPLIT_REJECT message. $C_5$, upon accepting the split request, is not allowed to slow down, and should wait for subsequent command from the leader. If $C_5$ slows down it causes its followers ($C_6$ and $C_7$) to change to the collision avoidance mode to prevent rear-end collision; this is not desirable, and should be avoided.

**3. Split execution:** The platoon leader, $C_1$, sends a unicast CHANGE_PL to $C_5$, and makes it a free agent (Fig. 6a). Now we have two leaders: $C_1^B$, and $C_1^A$. $C_1^A$ sends a multicast CHANGE_PL to all it's followers behind $C_1^B$, and asks them to change platoon leader to $C_1^B$. $C_1^A$ reports split completion by sending SPLIT_DONE to the $C_1^B$. Now $C_1^B$ can slow down safely and maintain inter-platoon spacing (Fig. 6b).

The state machine for platoon split maneuver is shown in Fig. 7. If optimal platoon size is *optPlnSize*, perform-

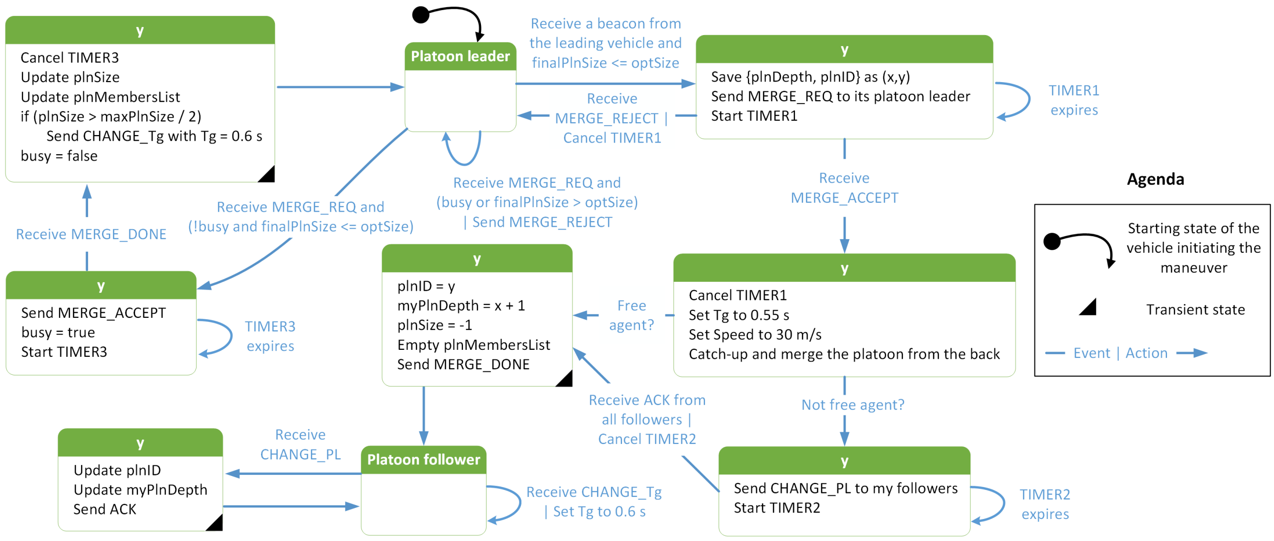

y
Cancel TIMER3
Update plnSize
Update plnMembersList
if (plnSize > maxPlnSize / 2)
Send CHANGE_Tg with Tg = 0.6 s
busy = false
Platoon leader
Receive a beacon from the leading vehicle and finalPlnSize <= optSize
y
Save {plnDepth, plnID} as (x,y)
Send MERGE_REQ to its platoon leader
Start TIMER1
TIMER1 expires
Receive MERGE_REJECT | Cancel TIMER1
Receive MERGE_REQ and (busy or finalPlnSize > optSize) | Send MERGE_REJECT
Receive MERGE_ACCEPT
Agenda
Starting state of the vehicle initiating the maneuver
Transient state
Event | Action
Receive MERGE_REQ and (!busy and finalPlnSize <= optSize)
Receive MERGE_DONE
y
Send MERGE_ACCEPT
busy = true
Start TIMER3
TIMER3 expires
y
plnID = y
myPlnDepth = x + 1
plnSize = -1
Empty plnMembersList
Send MERGE_DONE
Free agent?
y
Cancel TIMER1
Set Tg to 0.55 s
Set Speed to 30 m/s
Catch-up and merge the platoon from the back
Receive ACK from all followers | Cancel TIMER2
Not free agent?
y
Update plnID
Update myPlnDepth
Send ACK
Receive CHANGE_PL
Platoon follower
Receive CHANGE_Tg | Set Tg to 0.6 s
y
Send CHANGE_PL to my followers
Start TIMER2
TIMER2 expires

Figure 5: State machine for platoon merge.

ing a split maneuver on a platoon with size $plnSize(> optPlnSize)$ remains a platoon with size $optPlnSize$ and a platoon with size $plnSize - optPlnSize$. Other splitting positions are also feasible.

### 4.3. Lane Change Maneuver

Lane change is considered as a basic platoon maneuver in our protocol and is an important component for interactions with surrounding traffic. Vehicle entry uses lane change to enter into the reserved platooning lane while follower leave, and leader leave use lane change to exit from the reserved lane. In this paper, we consider lane change as a manual driving behavior and the default lane change model, LC2013 [10], in SUMO is adopted as the lateral control logic. Collaborative driving with lane change cooperation [30] is beyond the scope of this paper.

## 5. Platooning Scenarios

### 5.1. Platoon Leader Leave Maneuver

When a platoon leader needs to exit the platoon, it initiates a platoon leader leave maneuver (of course, if platoon size is 1 then platoon leader is free to leave). The platoon leader should ask a follower to take over as the leader of the platoon. The following steps show how the platoon leader leave maneuver is performed:

**1. Leave announcement:** VOTE_LEADER message is sent from the platoon leader to all its followers to vote on the new platoon leader role. Followers can vote on a new platoon leader by running a distributed leader selection algorithm. The new elected platoon leader uses ELECTED_LEADER message to announce this to the current platoon leader. At least one follower should respond, otherwise platoon leader resends VOTE_LEADER. For simplicity, we assume the second vehicle in the platoon will take over the leader role.

**2. Leave execution:** Platoon leader initiates a split maneuver to make itself a free agent in order to exit the platoon safely.

### 5.2. Follower Leave Maneuver

When a platoon follower needs to exit the platoon (for instance, the vehicle is approaching its destination), it initiates a leave maneuver, and then the control is relinquished to the driver in order to change lane. Only one vehicle is allowed to leave the platoon at a time. The basic idea here is to create enough space at the front (and rear) of the follower vehicle in order to make the lane change possible. Follower leave maneuver can be done using a sequence of split and merge maneuvers as follow:

**1. Leave request/response:** The follower vehicle $C_i$ notifies the platoon leader by sending a LEAVE_REQ. Platoon leader can send LEAVE_REJECT to reject this request or it can perform one or two split maneuvers in the next step to make $C_i$ a free agent.

**2. Leave execution:** If $C_i$ is the last vehicle in the platoon, one split maneuver is enough to make $C_i$ a free agent. Now $C_i$ is allowed to change lane and exit the platoon. On the other hand, if $C_i$ is not the last vehicle, two split plus one merge maneuvers are performed as depicted in Fig. 8.

### 5.3. Entry Maneuver

The entry maneuver should be safe and smooth. In the simplest case, steering the vehicle into the special lane can be left to the driver, and then the control is relinquished to the platoon management protocol. This approach relies on the driver to safely steer the vehicle into the special lane. A better approach would be to put this burden on the platoon management protocol, and the driver can simply trigger the entry maneuver by pushing a button. This will be a much safer approach, since the protocol has a better view of the surroundings, and can decides when will be a

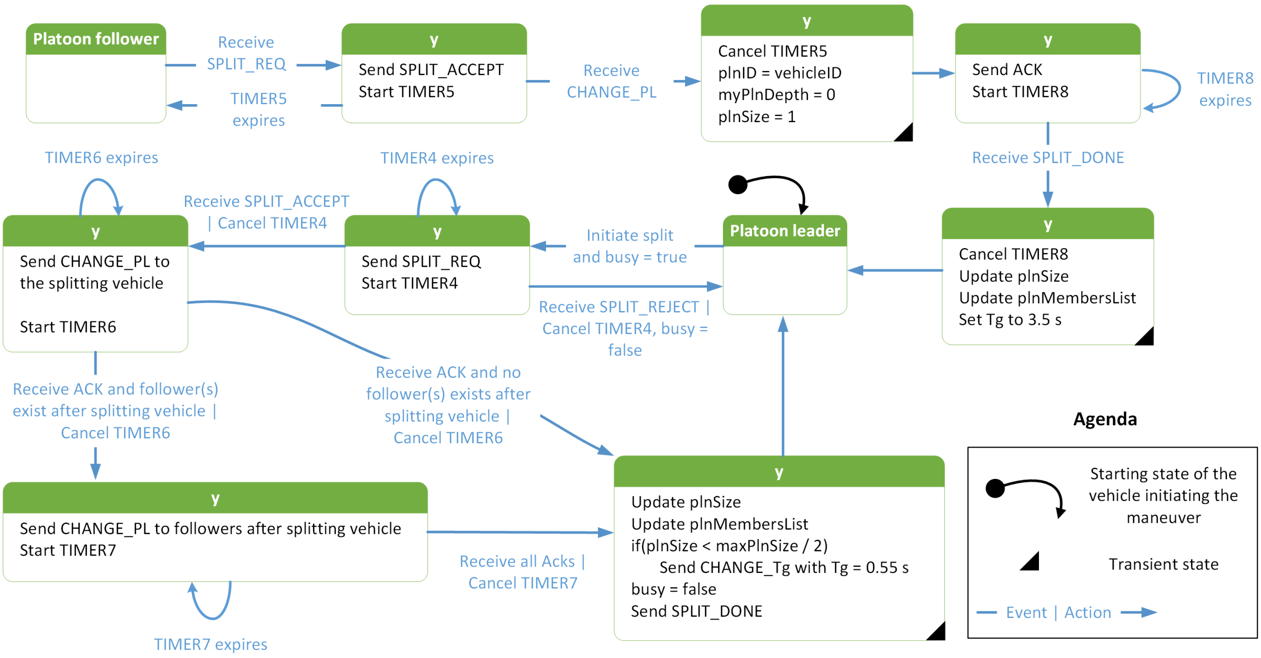

Platoon follower
Receive SPLIT_REQ
TIMER5 expires
y
Send SPLIT_ACCEPT
Start TIMER5
Receive CHANGE_PL
y
Cancel TIMER5
plnID = vehicleID
myPlnDepth = 0
plnSize = 1
y
Send ACK
Start TIMER8
TIMER8 expires
Receive SPLIT_DONE
TIMER6 expires
TIMER4 expires
Receive SPLIT_ACCEPT | Cancel TIMER4
y
Send CHANGE_PL to the splitting vehicle
Start TIMER6
y
Send SPLIT_REQ
Start TIMER4
Initiate split and busy = true
Platoon leader
Receive SPLIT_REJECT | Cancel TIMER4, busy = false
y
Cancel TIMER8
Update plnSize
Update plnMembersList
Set Tg to 3.5 s
Receive ACK and follower(s) exist after splitting vehicle | Cancel TIMER6
Receive ACK and no follower(s) exists after splitting vehicle | Cancel TIMER6
y
Send CHANGE_PL to followers after splitting vehicle
Start TIMER7
TIMER7 expires
Receive all Acks | Cancel TIMER7
y
Update plnSize
Update plnMembersList
if(plnSize < maxPlnSize / 2)
Send CHANGE_Tg with Tg = 0.55 s
busy = false
Send SPLIT_DONE
Agenda
Starting state of the vehicle initiating the maneuver
Transient state
Event | Action

Figure 7: State machine for platoon split.

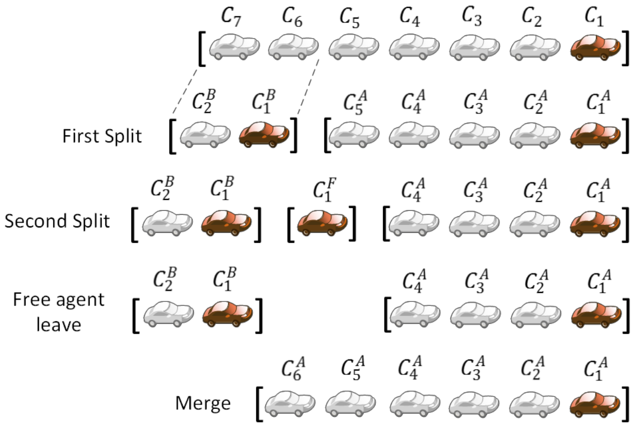

$C_7$ $C_6$ $C_5$ $C_4$ $C_3$ $C_2$ $C_1$
First Split
$C_2^B$ $C_1^B$ $C_5^A$ $C_4^A$ $C_3^A$ $C_2^A$ $C_1^A$
Second Split
$C_2^B$ $C_1^B$ $C_1^F$ $C_4^A$ $C_3^A$ $C_2^A$ $C_1^A$
Free agent leave
$C_2^B$ $C_1^B$ $C_4^A$ $C_3^A$ $C_2^A$ $C_1^A$
Merge
$C_6^A$ $C_5^A$ $C_4^A$ $C_3^A$ $C_2^A$ $C_1^A$

Figure 8: Following vehicle $C_5$ decides to leave the platoon (leaders are shown in red). Platoon leader $C_1$ issues the first split maneuver to split the platoon after $C_5$. Then it issues the second split maneuver to make $C_5^A$ a free agent. $C_1^F$ exits, and the two platoons are merged afterward, and the gap is closed.

good point to perform the entry maneuver. We will use the second approach, and show how the entry maneuver can be performed for vehicle $C_5$ in Fig. 3 that is driving on lane 0:

**1. Selecting a target platoon:** The first step is finding a target platoon to join. Platoon size of the target platoon is less than the maximum allowed platoon size, otherwise a new join request will be rejected by the platoon leader. A list of target platoons, sorted by some preferences, can be presented (by the RSU) to the driver to choose from. Two possible criteria for choosing a platoon are *route-based* and *cost-based*. In a route-based selection, the driver joins a platoon that is traveling on the route that she has selected for her trip. Joining a platoon that is traveling on the same route leads to fewer maneuvers. In cost-based, selection will be adopted where certain pricing may be applied for joining a platoon.

**2. Lane change:** $C_5$ starts listening to beacons to check if the target platoon is approaching on lane 1. Then it calculates the distance to the platoon leader, and measures the estimated arrival time $\xi$. If $\xi$ is big enough, then the platoon management protocol will steer $C_5$ into lane 1. $C_5$ then switches to CACC mode with intra-platoon gap $T_g = 3.5s$ and acts as a free agent. Subsequently, $C_5$ can merge into the target platoon.

Entry to the target platoon can be done from behind or side. Entry from side allows grouping of vehicles according to their destination which maximizes the distance that platoons stay intact [31, 32]. A high-rate of vehicles entering and exiting a platoon may cause vehicles to drive further apart, and decreases efficiency and safety. On the other hand, in mixed traffic platooning, side entry can be used to sort vehicles in order of their dynamical ability by putting the dynamically worst vehicle at the end of the platoon [33]. In this paper, we only focus on entry from behind.

## 6. Simulation Study

To study the performance of our platoon management protocol under realistic wireless communication and traffic flow scenarios, we used our simulator VENTOS (VEhicular NeTwork Open Simulator) [9]. VENTOS is an integrated

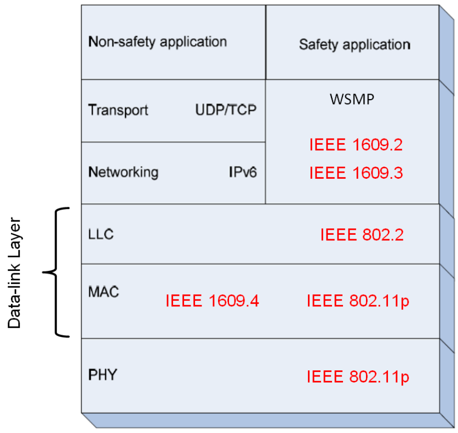


Figure 9: DSRC/WAVE network protocol stack [39] implemented in each VANET-enabled vehicle. WSMP header is attached to each beacon and platooning micro-command messages, and the resulting packet is sent directly to data-link layer.

simulator that is made up of many different modules, including Simulation of Urban Mobility (SUMO) and OMNET++/Veins. SUMO is an open source, microscopic, continuous-space, discrete-time road traffic simulator developed by Institute of Transportation Systems at the German Aerospace Center [34] and adopted as our traffic simulator. OMNET++ [35] is an open-source, component-based simulation package and captures the wireless communication simulation in VENTOS. IEEE 802.11p protocol, the standard protocol adopted for V2V communication in Veins (Vehicles in Network Simulation) framework [36], is used for wireless communication between CACC vehicles. Integration of SUMO and OMNET++ has been used in many previous researches and is getting more popular [37, 20, 7].

We use Wave Short Message Protocol (WSMP) [38] to carry beacon and micro-command messages on control channel (CCH), and the resulting message is directly sent to data-link layer as shown in Fig. 9. We use continuous channel access based on IEEE 1609.4 and always stay on CCH. Channel frequency is 5.89 GHz with data rate of 18 Mbps and transmission power of 10 mW.

### *6.1. Simulation Setting*

Simulation is done in a long straight two-lane highway with the speed limit of 30 m/s ($\approx$ 67 mph), and unidirectional traffic flow. Leftmost lane (lane 1) is dedicated to platooned vehicles and lane 0 is used for non-platooned vehicles. We assume vehicles are homogeneous and penetration of platooned-enabled vehicles (ratio of platooned-enabled vehicles to the vehicle population) is 100 percent. The highway starts with zero traffic, and the vehicles are inserted randomly into lane 0; Vehicles arrival follows a Poisson distribution with an average rate of $\lambda = 2000$ veh/h which is chosen to be less than highway throughput of 2400 veh/h under default settings. The default platooning settings are listed in Table 1.

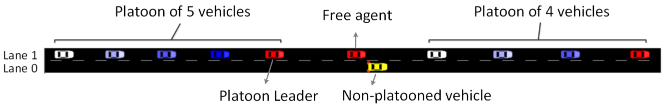


Figure 10: Snapshot of the simulation. Lane 1 is CACC specific lane that is reserved for platooned vehicles. Three platoons with size of 4, 1 (free agent), and 5 are traveling on lane 1.

As all vehicles are platooned-enabled, each vehicle on lane 0 changes lane to lane 1 at some point by performing an entry maneuver and then can participate in different platooning maneuvers. Platooned vehicles stay in lane 1 as long as they are part of a platoon, and upon leaving, they change to lane 0. Fig. 10 illustrates a snapshot of the simulation with three platoons in lane 1 including a platoon of 5 vehicles, a free agent, and a platoon of 4 vehicles. Platoon leader and its followers are shown with red and blue color respectively. As we go farther from the platoon leader, the blue color fades to show the platoon depth. Non-platooned vehicles in lane 0 are shown with yellow color.

### *6.2. Platooning Behavior of CACC Vehicle Stream*

We investigate vehicle string stability performance during split and merge maneuvers in a 10-vehicle platoon that is traveling with speed of 20 m/s. Fig. 11 shows the speed profile of vehicles in a split into two 5-vehicle platoons and then merge into one platoon. Split maneuver starts at time t=73.1s (mark 1), and Veh6 changes its desired time-gap from $T_g = 0.55s$ to $T_p = 3.5s$. As a result, Veh6 slows down to enlarge the space-gap and then accelerates to maintain $T_p = 3.5s$ with same speed of Veh5. Now we have two platoons with size 5, traveling with 20 m/s (mark 2). Subsequently, the two platoons start to merge at time t=118s (mark 3). Veh6 changes its time-gap setting to $T_g = 0.55s$ in order to speed up and merge with the front platoon. Eventually, a single platoon of 10 vehicles are formed (mark 4).

Fig. 12 illustrates how split and merge maneuvers are performed from another perspective. Position of the first vehicle in the stream (Veh1) is used as a reference point (dashed blue line) to measure the distance of vehicles to Veh1. The figure clearly shows how the gap between Veh6 and Veh5 slowly opens in split (mark 1) and slowly closes in merge (mark 2). Inter-platoon distance of 72 meters (mark 3) and intra-platoon distance of 14 meters (mark 4) are also visible.

After change of *optimal platoon size* (*optPlnSize*) in a road section, platoon management protocol attempts to meet this new platoon size as well as possible by performing a sequence of maneuvers as demonstrated in Fig. 13. Here we have a 10-vehicle platoon traveling on a road section with *optPlnSize* of 10. At time t=73s, *optPlnSize* changes to 2 and the platoon management protocol performs a sequence of four splits to break the platoon into smaller parts. We only allow one maneuver at a time,

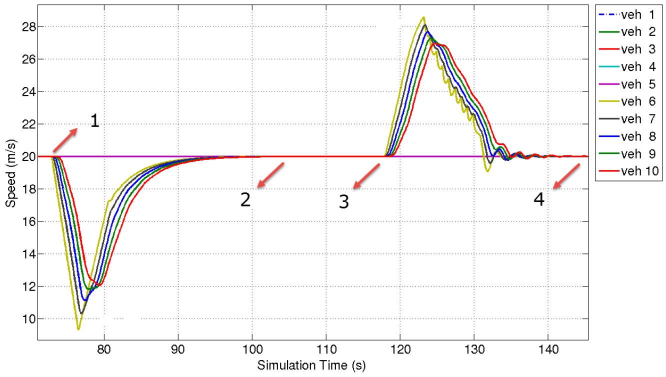


Figure 11: Speed profiles of platoon members during splitting a 10-vehicle platoon into two 5-vehicle platoons with Veh6 as the splitting vehicle starting at point marked 1, and merging of two 5-vehicle platoons starting at point marked 3. Speed profiles of the first 5 vehicles overlap on 20 m/s line.

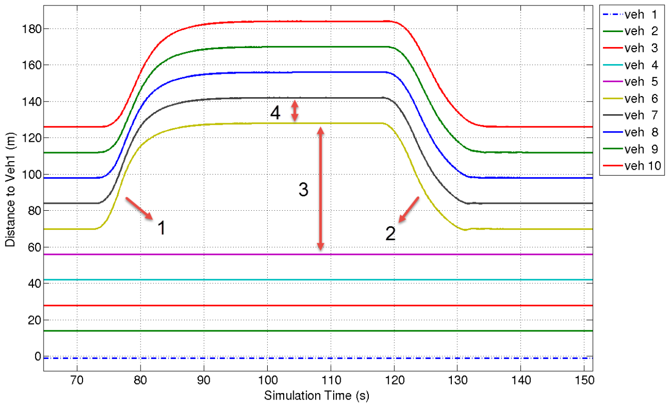


Figure 12: Distance of vehicles to the first vehicle in the stream (Veh1) during splitting a 10-vehicle platoon into two 5-vehicle platoons (with Veh6 as the splitting vehicle), and merging of two 5-vehicle platoons. Opening gap in split (mark 1), closing gap in merge (mark 2), inter-platoon distance (mark 3), and intra-platoon distance (mark 4) are shown.

thus these four splits are performed sequentially one after the other (starting point of each split is marked with 1-4). At t=130s, the *optPlnSize* is changed back to 10, and four merges are performed (starting point of each merge is marked with 5-8).

Fig. 14 shows the speed profiles of two 5-vehicle platoons that are traveling with speed of 20 m/s and the platoon leader of the front platoon (Veh1) is following the split/merge trajectory of Veh6 in Fig. 11. Speed profiles show that CACC control design can ensure string stability of platooning with small time-gap within platoons. Meanwhile, relatively large time-gap $T_p$ improves the stability of platooning by damping disturbances between platoons that enhances the stability of the whole traffic streams. Breaking long vehicle stream into smaller platoons not only ensures a better traffic flow stability, but also provides better safety and system robustness. This is due to the fact that collisions can be easily limited within one platoon, preventing chain collision between platoons.

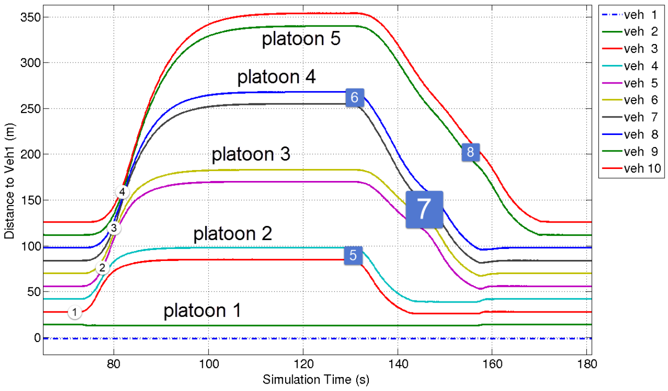


Figure 13: Effect of changing *optPlnSize* at t=73s from 10 to 2 on a 10-vehicle platoon. The platoon management protocol attempts to meet the new platoon size by performing four splits one after the other (as marked from 1 to 4). At t=130s, the *optPlnSize* is changed back to 10, and four merges (marked from 5 to 8) are performed to form a 10-vehicle platoon.

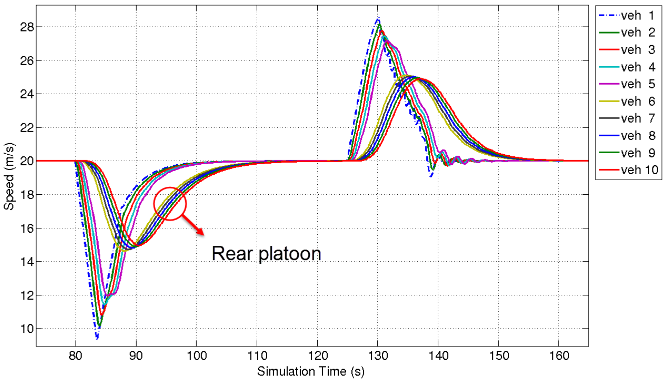


Figure 14: Speed profile of two 5-vehicle platoons with platoon leader of the front platoon (Veh1) follows a split/merge trajectory. This disturbance propagates through the vehicle stream.

Our simulation results demonstrate that CACC vehicles can maintain stable and smooth operations (in terms of speeds and inter-vehicle gaps) by following our platooning protocol.

### 6.3. Verification on throughput

Assuming a steady CACC platooning with platoon size of $N$ and speed of $V$ (m/s), inter-platoon spacing of $T_p$ (s), intra-platoon spacing of $T_g$ (s), vehicle length of $L_v$ (m), and minimum gap of $G_{min}$ (m), we can estimate the hourly throughput $Q$ (veh/h) of CACC specific lane by equation (1) adopted from [11]. Fig. 15 shows the estimated hourly throughput with different platoon sizes and intra-platoon spacing under default settings ($V = 20$ m/s, $T_p = 3.5$ s, $L_v = 5$ m, $G_{min} = 2$ m).

$$Q = \frac{VN}{VT_g(N-1) + VT_p + N(L_v + G_{min})} \times 3600 \quad (8)$$

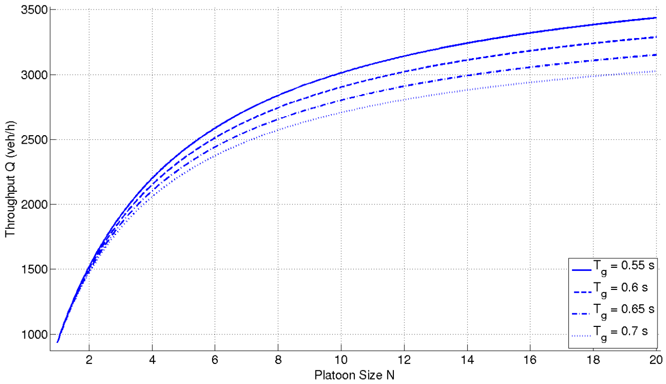


Figure 15: Throughput with different platoon size and intra-platoon time-gap setting under default traffic condition.

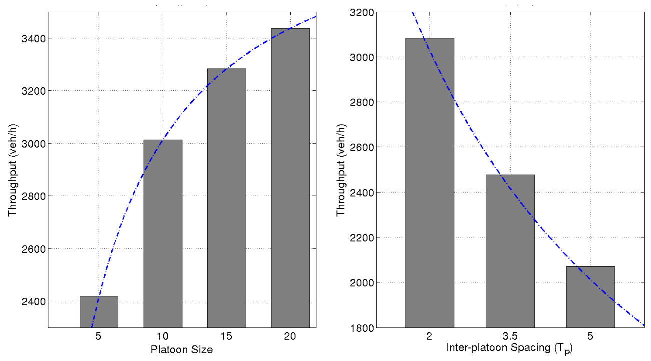


Figure 16: Effect of platoon size and $T_P$ on traffic throughput. The dashed blue lines show the estimated throughput.

Since our CACC control design and platooning management ensure high stability of CACC vehicle streams, simulated throughput is very close to the theoretical estimation. The impact of platoon size and inter-platoon time-gap setting $T_p$ on throughput is illustrated in Fig.16. As platoon size increases, traffic throughput also increases until a maximum traffic flow volume also known as *saturation flow* is reached. In other words, increasing the platoon size may increase throughput, but the enhancement is less when platoon size is large. Moreover, the relation between $T_p$ and throughput is linear. Based on this figure, platooning with platoon size of 10 and inter-platoon time-gap $T_p = 3.5s$ provides a good throughput as well as stability.

### 6.4. Duration of platooning maneuvers

To measure the merge/split duration, we repeat a total of 9 scenarios in which a 10-vehicle platoon splits in different positions into two smaller platoons, and then they merge again as a 10-vehicle platoon. Fig. 17 illustrates the average duration time of merge, split, and leaving maneuvers in a 10-vehicle platoon traveling with speed of 20 m/s under default platooning settings. Space-gap between two platoons is around 72 meters, maximum catch up speed is 30 m/s, and maximum acceleration and deceleration are 2 $m/s^2$ and -3 $m/s^2$, respectively.

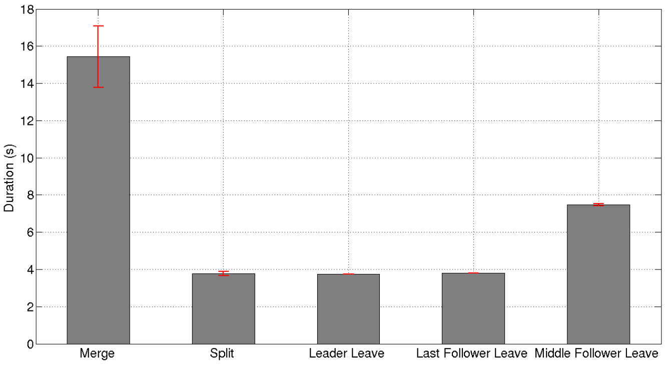


Figure 17: Average maneuver duration for merge, split, and leave (leader leave, last follower leave, and middle follower leave).

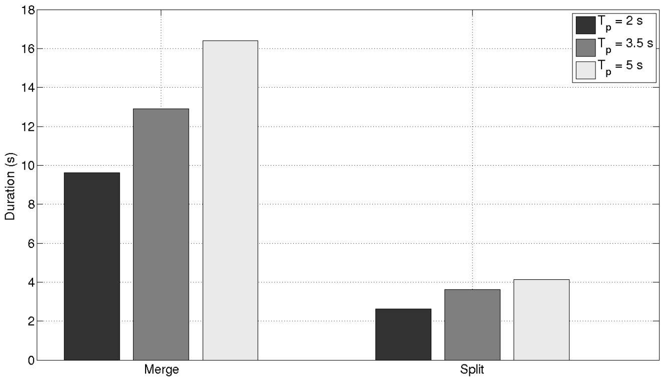


Figure 18: Effect of $T_P$ on merge and split duration.

We differentiate between three types of leaving: platoon leader leave, middle follower leave, and last follower leave. Middle follower leave takes almost twice as much as leader or middle follower leave since two splits are required in order to make enough front/rear space for the middle follower. All vehicles are assumed to be homogeneous and communication channel is perfect. The confidence intervals are small since the wireless communications has small interference resulting in small and almost deterministic delays.

Increasing inter-platoon time-gap not only decreases the throughput, but also increases the duration of split and merge maneuvers, as shown in Fig. 18. Increasing $T_p$ leads to bigger space-gaps between platoons; therefore, vehicles needs more time to fill this gap (in merge) or create one (in split). Large $T_p$ ensures a better stability, safety, and easier lane change for entry maneuver. $T_p$ should be chosen to balance different aspects, including the frequency of splitting and merging maneuvers.

### 6.5. Impact of Communication Failure

Wireless communication failure can affect the performance of CACC vehicles and lead to string instability or even rear-end collision. It might also have significant impact on the platoon management protocol that relies on wireless communication to exchange different messages between vehicles. Detailed study of communication failure such as stability analysis and packet loss tolerance are beyond the scope of this paper. In this section we will only show the impact of communication failure on our CACC platooning, and how the system can recover itself from such failures.

**Effect of communication failure on CACC controller:** Our designed CACC controller can detect communication loss by simply monitoring the timestamp of the received data. Failure to receive beacon for $\gamma$ seconds triggers the communication failure flag in CACC controller. The value of $\gamma$ depends on many parameters including beaconing rate and needs to be studied further to measure the packet loss tolerance. In our simulation study, we consider the most conservative case and set $\gamma$ to be equal to beaconing rate ($= 0.1$s). Hence, missing only one beacon message in each simulation time step triggers the communication failure flag to be set.

Downgrading from CACC to ACC is a feasible solution that diminishes the impact of communication failure from a rear-end collision to reduction in CACC performance. Fig. 19 illustrates the space-gap between CACC vehicles in a 10-vehicle platoon that are traveling with speed of 20 m/s, and intra-platoon spacing of 16 m (mark 1). Platoon leader (Veh1) slows down to 5 m/s and as a result, intra-platoon spacing decreases to 5.5 m (mark 2). At t=308s, all wireless communications are disrupted (mark 3) which mimic the situation of noisy channel or security attack by an adversary. Failure to receive beacons, CACC vehicles downgrade to ACC mode with larger time-gap and controller delay settings. As a result they slow down to increase the space-gap (mark 4), and then resume at 5 m/s. Platoon leader speeds up again to 20 m/s, but this time the space-gap is 26 m (mark 5). Reaction of followers to speed changes becomes relatively slower (mark 6b) than in CACC mode (mark 6a).

**Effect of communication failure on platoon management protocol:** Communication failure can also affect maneuver operation significantly, and it is crucial to design a platoon management protocol that is resilient to packet losses, and be able to take correcting actions or abort the maneuver safely. As an example, we illustrate how 'platoon leader leave' maneuver reacts under communication failure in Fig. 20. The platoon leader initiates a leave maneuver (as detailed in Section 5.1) at t=73s but due to communication failure, no response is received from followers to take over the platoon leader role. As a result, the platoon leader breaks-up the platoon by using a DISSOLVE micro-command.

## 7. Conclusion

In this paper, a high-level design of a platoon management protocol with three basic maneuvers: merge, split, and lane-change is proposed that is based on Vehicular Ad-hoc Network (VANET) and CACC vehicles. A set

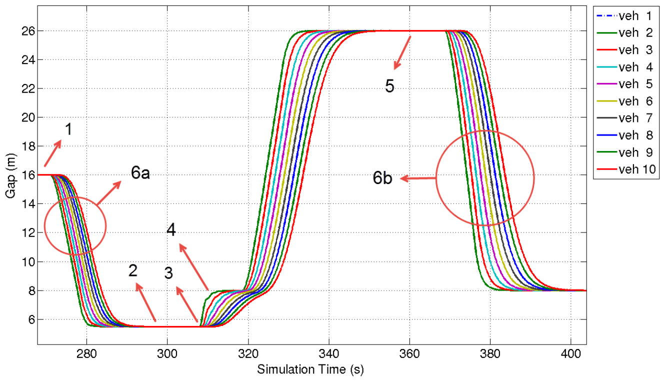


Figure 19: Performance of CACC vehicle stream under wireless communication failure after downgrading to ACC. *Mark 1:* a platoon of 10 CACC vehicles is traveling with speed of 20 m/s, and intra-platoon spacing of 16 m. *Mark 2:* platoon leader (Veh1) slows down to 5 m/s and as a result, intra-platoon spacing decreases to 5.5 m. *Mark 3:* all wireless communications are disrupted, and CACC vehicles switch to ACC mode. *Mark 4:* vehicles slow down to increase the space-gap. *Mark 5:* platoon leader speeds up again to 20 m/s, but this time the space-gap is 26 m. *Mark 6a,6b:* After switching to ACC, reaction of followers to speed changes become relatively slower.

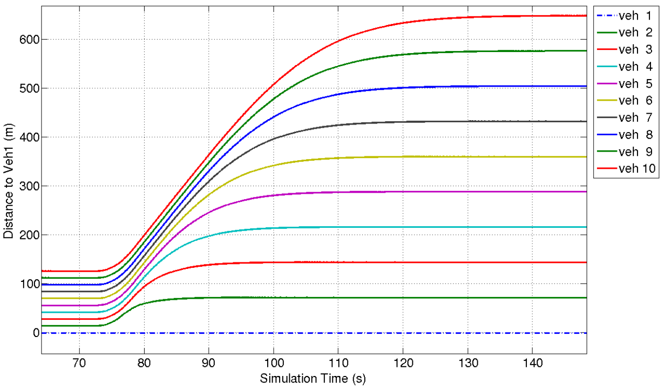


Figure 20: Impact of communication failure on platoon leader leave maneuver. The platoon leader dissolves the platoon into nine 1-vehicle platoons.

of micro-commands exchanged between vehicles leveraging IEEE 802.11p are used to accomplish the basic maneuvers, and the protocol operation is described in details using various finite state machines. We implement this protocol in VENTOS, an integrated simulation platform based on SUMO and OMNET++, and show the effectiveness of our platooning protocol. Moreover, a comprehensive CACC longitudinal control system is designed and implemented in SUMO to better model the real vehicle dynamics. Through simulation study, we show that our protocol ensures traffic flow stability and theoretical throughput is maintained. Furthermore, our CACC controller as well as platoon management protocol can react to communication loss by message retransmission or downgrading to ACC mode, but more detailed study is needed. Our designed protocol can be easily extended to incorporate different strategies in collaborative driving and intelligent highway systems.

## References

[1] M. El-Zaher, B. Dafflon, F. Gechter, J.-M. Contet, Vehicle platoon control with multi-configuration ability, Procedia Computer Science 9 (2012) 1503–1512.

[2] M. Michaelian, F. Browand, Field experiments demonstrate fuel savings for close-following, California PATH Program, Institute of Transportation Studies, University of California at Berkeley, 2000.

[3] A. A. Alam, A. Gattami, K. H. Johansson, An experimental study on the fuel reduction potential of heavy duty vehicle platooning, in: Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference, IEEE, 2010, pp. 306–311.

[4] F. Browand, J. McArthur, C. Radovich, Fuel saving achieved in the field test of two tandem trucks, California PATH research report.

[5] S. Tsugawa, S. Kato, K. Aoki, An automated truck platoon for energy saving, in: Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference, IEEE, 2011, pp. 4109–4114.

[6] M. Parent, G. Gallais, Intelligent transportation in cities with cts, in: Proceedings of the IEEE 5th International Conference on Intelligent Transportation Systems, 2002, IEEE, 2002, pp. 826–830.

[7] D. Jia, K. Lu, J. Wang, On the network connectivity of platoon-based vehicular cyber-physical systems, Transportation Research Part C: Emerging Technologies.

[8] C. Bergenhem, S. Shladover, E. Coelingh, C. Englund, S. Tsugawa, Overview of platooning systems, in: Proceedings of the 19th ITS World Congress, Oct 22-26, Vienna, Austria (2012), 2012.

[9] M. Amoozadeh, Vehicular Network Open Simulator (VENTOS), More details can be found at this address: http://rubinet.ece.ucdavis.edu/projects/ventos.

[10] J. Erdmann, Lane-Changing Model in SUMO, proceedings of SUMO2014 24 (2014) 77–88.

[11] P. Varaiya, Smart cars on smart roads: problems of control, IEEE Transactions on Automatic Control 38 (2) (1993) 195–207.

[12] H.-S. Tan, R. Rajamani, W.-B. Zhang, Demonstration of an automated highway platoon system, in: American Control Conference, 1998, Vol. 3, IEEE, 1998, pp. 1823–1827.

[13] S. Tsugawa, S. Kato, T. Matsui, H. Naganawa, H. Fujii, An architecture for cooperative driving of automated vehicles, in: Intelligent Transportation Systems, 2000, IEEE, 2000, pp. 422–427.

[14] S. Kato, S. Tsugawa, K. Tokuda, T. Matsui, H. Fujii, Vehicle control algorithms for cooperative driving with automated vehicles and intervehicle communications, IEEE Transactions on Intelligent Transportation Systems 3 (3) (2002) 155–161.

[15] S. Hallé, F. Gilbert, J. Laumonier, B. Chaib-draa, Architectures for collaborative driving vehicles: From a review to a proposal, Citeseer, 2003.

[16] P. Fernandes, U. Nunes, Platooning with ivc-enabled autonomous vehicles: Strategies to mitigate communication delays, improve safety and traffic flow, IEEE Transactions on Intelligent Transportation Systems 13 (1) (2012) 91–106.

[17] F. Michaud, P. Lepage, P. Frenette, D. Letourneau, N. Gaubert, Coordinated maneuvering of automated vehicles in platoons, IEEE Transactions on Intelligent Transportation Systems 7 (4) (2006) 437–447.

[18] S. Hallé, J. Laumonier, B. Chaib-draa, A decentralized approach to collaborative driving coordination, in: The 7th International IEEE Conference on Intelligent Transportation Systems, 2004, IEEE, 2004, pp. 453–458.

[19] S. Hallé, B. Chaib-draa, A collaborative driving system based on multiagent modelling and simulations, Transportation Research Part C: Emerging Technologies 13 (4) (2005) 320–345.

[20] M. Segata, F. Dressler, R. Lo Cigno, M. Gerla, A simulation tool for automated platooning in mixed highway scenarios, in: Proceedings of the 18th annual international conference on Mobile computing and networking, ACM, 2012, pp. 389–392.

[21] M. Segata, S. Joerer, B. Bloessl, C. Sommer, F. Dressler, R. Lo Cigno, PLEXE: A Platooning Extension for Veins, in: 6th IEEE Vehicular Networking Conference (VNC 2014), IEEE, Paderborn, Germany, 2014.

[22] T. Robinson, E. Chan, E. Coelingh, Operating platoons on public motorways: an introduction to the sartre platooning programme, in: 17th World Congress on Intelligent Transport Systems, 2010, pp. 1–11.

[23] F. Bu, H.-S. Tan, J. Huang, Design and field testing of a cooperative adaptive cruise control system, in: American Control Conference (ACC), 2010, IEEE, 2010, pp. 4616–4621.

[24] H. Hartenstein, K. P. Laberteaux, A tutorial survey on vehicular ad hoc networks, Communications Magazine, IEEE 46 (6) (2008) 164–171.

[25] P. Fernandes, U. Nunes, Platooning with dsrc-based ivc-enabled autonomous vehicles: Adding infrared communications for ivc reliability improvement, in: Intelligent Vehicles Symposium (IV), 2012 IEEE, IEEE, 2012, pp. 517–522.

[26] J. Kjellberg, Implementing control algorithms for platooning based on v2v communication, Master Thesis.

[27] A. Uchikawa, R. Hatori, T. Kuroki, H. Shigeno, Filter multicast: a dynamic platooning management method, in: Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE, IEEE, 2010, pp. 1–5.

[28] S. Kanda, M. Suzuki, R. Harada, H. Shigeno, A multicast-based cooperative communication method for platoon management (poster), in: Vehicular Networking Conference (VNC), 2011 IEEE, IEEE, 2011, pp. 185–192.

[29] R. Rajamani, Vehicle dynamics and control, Springer, 2011.

[30] R. Rajamani, H.-S. Tan, B. K. Law, W.-B. Zhang, Demonstration of integrated longitudinal and lateral control for the operation of automated vehicles in platoons, IEEE Transactions on Control Systems Technology 8 (4) (2000) 695–708.

[31] R. Hall, C. Chin, Vehicle sorting for platoon formation: Impacts on highway entry and throughput, Transportation Research Part C: Emerging Technologies 13 (5) (2005) 405–420.

[32] T.-S. Dao, C. M. Clark, J. P. Huissoon, Optimized lane assignment using inter-vehicle communication, in: Intelligent Vehicles Symposium, 2007 IEEE, IEEE, 2007, pp. 1217–1222.

[33] A. Mihály, P. Gáspár, Control of platoons containing diverse vehicles with the consideration of delays and disturbances, Transportation Engineering 40 (1) (2013) 21–26.

[34] D. Krajzewicz, G. Hertkorn, C. Rössel, P. Wagner, SUMO (Simulation of Urban MObility), in: Proc. of the 4th Middle East Symposium on Simulation and Modelling, 2002, pp. 183–187.

[35] A. Varga, et al., The omnet++ discrete event simulation system, in: Proceedings of the European Simulation Multiconference (ESM2001), Vol. 9, sn, 2001, p. 185.

[36] C. Sommer, R. German, F. Dressler, Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis, IEEE Transactions on Mobile Computing 10 (1) (2011) 3–15. doi:10.1109/TMC.2010.133.

[37] M. Amoozadeh, Certificate revocation list distribution in vehicular communication systems, Master Thesis, LCN Lab, KTH University, 2012.

[38] Intelligent Transportation Systems Committee and others, IEEE Trial-Use Standard for Wireless Access in Vehicular Environments (WAVE)Networking Services, IEEE Std (2007) 1609–3.

[39] Y. J. Li, An overview of the DSRC/WAVE technology, in: Quality, Reliability, Security and Robustness in Heterogeneous Networks, Springer, 2012, pp. 544–558.

[40] A. S. Le Yi Wang, G. Yin, A. Pandya, H. Zhang, Coordinated vehicle platoon control: weighted and constrained consensus and communication network topologies, Proceedings of the Conference on Decision and Control (CDC), 2012 4057–4062.

[41] L. Hobert, A study on platoon formations and reliable communication in vehicle platoons, Master Thesis, University of Twente.

## Appendix A. Platoon Management: Variables and Micro-Commands

### *Appendix A.1. Variables*

Platoon management protocol in each platoon-enabled vehicle keeps track of different variables such as: *vehicle id*, *platoon id*, *platoon depth*, *platoon size*, and *platoon members*. These variables are managed differently based on the vehicle type as shown in Table A.2. Vehicle Id is an identifier that is used to uniquely specify a vehicle, and can be regarded as the vehicle registration plate. In simulation environment like OMNET++, vehicles can use module id for this purpose. OMNET++ guarantees that the assigned ids are unique. Platoon members variable is a list that contains the vehicle ids of all platoon members. Only the platoon leader keeps track of the platoon size, and platoon members list. In maneuvers like *platoon leader leave*, old platoon leader should pass these variables to the new platoon leader.

### *Appendix A.2. Micro-Commands*

Table A.3 shows the defined micro-commands. Each micro-command is using a specific application-layer transmission type. *Unicast transmission* shows a one-to-one relation between sender and receiver. *Multicast transmission* shows a one-to-many relation where sender is sending to a group of receivers. For example, CHANGE_Tg micro-command which is sent from the platoon leader to all following vehicles is using multicast transmission. Lastly, *broadcast transmission* shows one-to-all relation where the micro-command is received by all nearby entities, and can be used in inter-platoon communication. We will describe some of the micro-commands in the following.

**MERGE_REQ:** This micro-command is a unicast message sent from the platoon leader of rear platoon to the platoon leader of front platoon, asking to merge. Platoon leader of the rear platoon should include enough information about its platoon configuration (such as platoon size, followers type, etc) in order to assist the front platoon leader in accepting/rejecting the merge request.

**MERGE_ACCEPT/REJECT:** These are unicast messages sent from the platoon leader in response to the MERGE_REQ. The platoon leader can accept the merge request by sending MERGE_ACCEPT, or it might reject the merge request by sending MERGE_REJECT. Platoon leader's decision can be based on different parameters, such as the current platoon size. If the front platoon is busy performing a maneuver, or the *final platoon size* > *optimal platoon size*, then platoon leader can simply reject the join request.

**MERGE_DONE:** Having merged into the front platoon, the old platoon leader of the rear platoon sends this unicast message to the platoon leader. Upon receiving MERGE_DONE, the platoon leader updates platoon size, followers list, and un-sets the busy flag.

**CHANGE_PL:** Platoon leader uses this message to announce the new platoon leader to its follower(s). This can be triggered by different events, such as, merging, platoon split, etc. Each follower, upon receiving this message, updates the platoon id and platoon depth. This means that CHANGE_Tg message should convey enough information so that followers can update their internal states accordingly.

**CHANGE_Tg:** This is a unicast message sent from the platoon leader to its follower(s), asking to change the time-gap. In order to have best control of platoon to ensure large platoon size and good stability, we may change intra-platoon time-gap setting based on graded platoon size. Moreover, time-gap setting is dependent on the weather, road condition, platoon traveling speed, terrain composition (uphill or downhill), and road curvatures [40].

**VOTE_LEADER:** This is a multicast message sent from the platoon leader to all its followers, voting on the new platoon leader role. *Platoon leader leave* can trigger this micro-command. This message contains the current platoon configuration. All following vehicles receive this message and can vote on a new platoon leader based on a distributed leader selection algorithm.

**ELECTED_LEADER:** The new elected platoon leader, announces itself to the current platoon leader using this unicast message. Subsequently, the platoon leader asks all its followers to change the platoon leader to the new one using CHANGE_PL message.

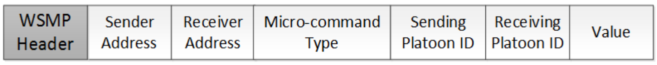
**DISSOLVE:** This is a multicast message from platoon leader to all followers in order to break-up the platoon. For instance, in the *platoon leader leave* maneuver, if no follower is willing to take over the new platoon leader role, then the platoon can be dissolved. Emergency situations like urgent leave of the platoon leader might also trigger this message. After dissolution, platoon members acts as

| Variable | Non-platooned | Platoon Follower | Platoon Leader |
|---|---|---|---|
| **Vehicle Id** | Unique string | Unique string | Unique string |
| **Platoon Id** | X | Vehicle Id of the leader | Same as Vehicle Id |
| **Platoon depth** | X | $\geq 1$ | 0 |
| **Platoon size** | X | X | $\geq 1$ |
| **Platoon members** | X | X | Followers Vehicle Id |

Table A.2: Variables maintained by platoon management protocol in each platoon-enabled vehicle

| | Micro-command | Description | Sender / Receiver |
|---|---|---|---|
| 1 | MERGE_REQ (U) | Asking to merge into a platoon | Rear platoon leader to the front platoon leader |
| 2 | MERGE_ACCEPT (U) | Accepting merge request | Front platoon leader in reply to MERGE_REQ |
| 3 | MERGE_REJECT (U) | Rejecting merge request | Front platoon leader in reply to MERGE_REQ |
| 4 | MERGE_DONE (U) | Notifying the merge completion | Old rear platoon leader to the platoon leader |
| 5 | SPLIT_REQ (U) | Split the platoon into two parts | Platoon leader to a follower |
| 6 | SPLIT_ACCEPT (U) | Accepting split request | Follower to the platoon leader in reply to SPLIT_REQ |
| 7 | SPLIT_REJECT (U) | Rejecting split request | Follower to the platoon leader in reply to SPLIT_REQ |
| 8 | SPLIT_DONE (U) | Notifying the split completion | Original platoon leader to leader of second platoon |
| 9 | LEAVE_REQ (U) | Follower, asking to leave the platoon | A follower to the platoon leader |
| 10 | LEAVE_ACCEPT (U) | Accepting leave request | Platoon leader to the follower in reply to LEAVE_REQ |
| 11 | LEAVE_REJECT (U) | Rejecting leave request | Platoon leader to the follower in reply to LEAVE_REQ |
| 12 | VOTE_LEADER (M) | Voting on the new platoon leader role | Current platoon leader to all followers |
| 13 | ELECTED_LEADER (U) | Accepting the new platoon leader role | Elected vehicle to the current platoon leader |
| 14 | DISSOLVE (M) | Platoon break-up | Platoon leader to all its followers |
| 15 | CHANGE_PL (U or M) | Announcing the new platoon leader | Current platoon leader to all its followers |
| 16 | CHANGE_Tg (M) | Asking to change the time-gap setting | Platoon leader to all its followers |
| 17 | ACK (U) | Acknowledging the message reception | The receiving entity to the sending entity |

Table A.3: Micro-commands used in platoon management protocol. U: Unicast, M: Multicast, B: Broadcast



| WSMP Header | Sender Address | Receiver Address | Micro-command Type | Sending Platoon ID | Receiving Platoon ID | Value |
|---|---|---|---|---|---|---|

Figure A.21: Micro-command message format

a free agent, and can merge into other platoons or leave the special lane.

**ACK:** Unicast messages occur in request-reply pairs in which the sender transmits a request and waits for the reply. Lack of reply means either the request or the reply was lost during transmission, and the request is retransmitted. In multicast messages, all receiving vehicles, acknowledge the message reception using ACK. More efficient techniques such as *Cooperative Acknowledgments (COACK)* can also be used where another vehicle in the platoon can forward and deliver the acknowledgement to the sender [41].

All micro-commands use the same message format as depicted in Fig. A.21. Sending entity fills the sender address with its vehicle id. Receiver address can be a unicast or broadcast address or it might be a group address in multicast transmission. Micro-command type is an integer value that uniquely specifies a micro-command. Sending and receiving platoon ids add another dimension to the addressing and specify the exact platoon id that this message is sending from/going to. Data might be attached to a micro-command into the value field.