Received 6 December 2022; revised 22 February 2023 and 4 April 2023; accepted 22 April 2023. Date of publication 28 April 2023; date of current version 9 May 2023.

Digital Object Identifier 10.1109/OJITS.2023.3271608

PlatoonSAFE: An Integrated Simulation Tool for Evaluating Platoon Safety

SHAHRIAR HASAN[1] (Student Member, IEEE), JOSEBA GOROSPE[2], SVETLANA GIRS[1] (Member, IEEE), ARRATE ALONSO GÓMEZ[2], AND ELISABETH UHLEMANN[1] (Senior Member, IEEE)

[1]School of Innovation, Design and Engineering, Mälardalen University, 722 20 Västerås, Sweden

[2]Electronics and Computer Science Department, Mondragon Unibertsitatea, 20500 Arrasate, Spain

CORRESPONDING AUTHOR: S. HASAN (e-mail: shahriar.hasan@mdu.se)

This work was supported in part by the European Union's Horizon 2020 Research and Innovation Programme under the Marie Skłodowska-Curie Project ImmerSAFE under Grant 764951. The work of Joseba Gorospe and Arrate Alonso Gómez was supported by the Basque Government's Research Group Program under Grant IT1451-22 and through the Project AUTOEV@L under Grant KK-2021/00123. The work of Elisabeth Uhlemann was supported by the Swedish Foundation for Strategic Research through the Serendipity Project.

ABSTRACT Platooning is highly tractable for enabling fuel savings for autonomous and semi-autonomous cars and trucks. Safety concerns are one of the main impediments that need to be overcome before vehicle platoons can be deployed on ordinary roads despite their readily available technical feasibility. Simulation studies remain vital for evaluating platoon safety applications primarily due to the high cost of field tests. To this end, we present PlatoonSAFE, an open-source simulation tool that promotes the simulation studies of fault tolerance in platooning by enabling the monitoring of transient communication outages during runtime and assigning an appropriate performance level as a function of the instantaneous communication quality. In addition, PlatoonSAFE facilitates the simulation of several emergency braking strategies to evaluate their efficacy in transitioning a platoon to a fail-safe state. Furthermore, two Machine Learning (ML) models are integrated into PlatoonSAFE that can be employed as an onboard prediction tool in the platooning vehicles to facilitate online training of ML models and real-time prediction of communication, network, and traffic parameters. In this paper, we present the PlatoonSAFE structure, its features and implementation details, configuration parameters, and evaluation metrics required to evaluate the fault tolerance of platoon safety applications.

INDEX TERMS CACC, cooperative driving, connected vehicles, discrete event simulations, fail-operational, fail-safe, fault tolerance, machine learning, platoon, PLEXE, SUMO, Veins, V2V.

I. INTRODUCTION

CONNECTED and Automated Vehicles (CAVs) allow for coordination and connectivity between vehicles through Vehicle-to-Vehicle (V2V) and with infrastructure through Vehicle-to-Infrastructure (V2I) communications and employ advanced sensing technology to enhance safety, fuel efficiency, traffic flow, and other aspects of transportation. Cooperative Adaptive Cruise Control (CACC) is a technology used in CAVs that improves upon traditional sensor-based Adaptive Cruise Control (ACC) by incorporating V2V communications to regulate vehicle speed and facilitate longitudinal control. Platooning is another CAV technology that enables vehicles to form a flexible and reconfigurable road train for lateral and longitudinal control [1]. In contrast to standalone CACC technology, platooning vehicles require high coordination. Following Vehicles (FVs) in a platoon use V2V communications and onboard sensors to track the position, speed, acceleration, and steering angle of the Leading Vehicle (LV), maintaining short inter-vehicle gaps. The combination of CACC and platooning technologies in CAVs hold great potential for promoting a safer, cleaner, and more sustainable transportation system on the roads.

However, wireless communications, a key enabling technology in platooning applications, is often prone to transient

The review of this article was arranged by Associate Editor Jia Hu.

communication outages, which pose significant challenges in ensuring proper vehicle control and safety. Consequently, even if encountering transient communication outages, a platoon is required to remain *fail-operational*, i.e., certain critical functionalities should be retained while maintaining a nominal safety level instead of simply dissolving a platoon which is an option, e.g., if the vehicles are manually driven. In addition, the communication delays between vehicles in a platoon can be very high when in dense traffic scenarios [2], [3]. A platoon may also be required to perform emergency braking due to encountering a road hazard. In the event of a hazard, a platoon must be able to transition to a *fail-safe state* so that people, environment, and equipment remain unharmed. The fail-operational and fail-safe states in the presence of a *combination of faults* can be regarded as fault tolerance in platooning [4]. In order to evaluate platoon fault tolerance, the time-varying nature of wireless connectivity must be considered, which is caused by factors such as path loss, multipath fading, shadowing, signal attenuation, unbounded channel access delays in the IEEE 802.11p Medium Access Control (MAC) protocol [5], and more. Therefore, the assumptions such as constant communication delay [6], communication is either present or absent between two links [7], etc., cannot accurately model the realistic nature of wireless connectivity. Moreover, different vehicles in a platoon may experience different communication outages, and the experienced communication delays during cruising might also be retained during emergency braking. Therefore, platoon cruising and emergency braking are tightly coupled by the impacts of time-varying communication delays, and such inter-dependency should be considered for designing platoon fault tolerance. To this end, this paper presents a simulation tool facilitating realistic modeling of transient communication outages. Other factors, including vehicle dynamics, road conditions, neighboring traffic, inter-vehicle gaps, speed, deceleration rate, controller, and more, are also crucial for evaluating platoon safety alongside wireless connectivity.

Analyzing platoon safety using mathematical modeling or theoretical analysis often requires simplifying the wireless communication model as either a successful or failed connection, see, e.g., [7] and [8]. Further, Weber et al. report in their survey on Vehicular Ad hoc Networks (VANETs) simulators [9] that almost none of the well-known and open source VANETs simulators implement or facilitate the evaluation of fault tolerance mechanisms, with one exception, the technical report in [10]. This is a report by us in which we conceptualized the implementation and simulation of fault tolerance regimes. The report is now extended, and the part covering the fine-grained characterization of transient communication outages facilitating graceful performance degradation and upgradation to enable fault tolerance in platooning is presented in [2]. In addition, the part of the report covering the simulator is described in this paper. Moreover, Almeida et al. report that simulation studies of fault tolerance techniques in VANETs should consider a more integrated approach covering the overall network architecture and its impact on the system performance, which is nonexistent in the literature [11]. In addition, the prediction of different communication, network, and traffic parameters in VANETs through Machine Learning (ML) models can potentially provide the required Quality of Service (QoS) and reliability necessary for platoon safety applications. Several recent works propose implementing ML models in the base station of a centralized network, e.g., [12] and [13], or collecting data through simulations and training ML models offline for making predictions, e.g., [14]. However, in a decentralized network of vehicles, i.e., VANET, the network and mobility of the vehicles frequently change. This demands online and real-time predictions of communication, network status, and mobility parameters. To the best of our knowledge, no VANETs or platooning simulator is available in the community that enables evaluating fault tolerance regimes in the presence of transient communication outages. Moreover, the existing simulation platforms do not facilitate integrating ML models in the platooning vehicles as an onboard prediction tool or allow online ML model training using communications, network, or traffic parameter values to enable real-time predictions of, e.g., communication delays.

Moreover, Lai et al. [15] and Hu et al. [16] categorize the scientists who developed simulation platforms over the years for evaluation and validation of CACC vehicle strings into two groups: automotive engineers and transportation engineers. The Automotive Engineer Developed Platforms (AEDPs) mainly focus on vehicle dynamics and often simplify the possibility of large-scale and mixed traffic scenarios, e.g., [17], whereas the Traffic Engineer Developed Platforms (TEDPs) focus on realistic road traffic simulations and make simplified assumptions on vehicle dynamics, e.g., [18]. In addition, the AEDPs and TEDPs often make simplified assumptions about communication delays incurred by platooning or CACC vehicles, i.e., time-varying communication delays or transient communication outages are not considered, e.g., [6]. To this end, we would like to add another group of platforms to the set, namely Communication Engineer Developed Platforms (CEDPs), which mainly focus on the communication aspects of platooning or CACC vehicle strings for evaluation and validation purposes. One example of such CEDPs is Veins [19], a simulation platform for evaluating VANETs applications. In order to evaluate the performance of CACC vehicle strings, vehicle platoons, or in general, CAVs, an integrated approach to developing simulation platforms should be taken, considering realistic models of road traffic, vehicle dynamics, *and* wireless communication. The PlatoonSAFE simulation tool presented in this paper aims at providing the last piece of the puzzle towards an integrated simulator by combining AEDPs, TEDPs, and CEDPs.

In search of an existing open-source simulator that can potentially bridge between AEDPs, TEDPs, and CEDPs, we found the platooning simulator PLEXE [20] as a viable candidate, which allows realistic simulation of vehicle dynamics, road traffic, platoon control algorithms, and full-fledged network stack that is inherited from Veins. In this paper, we present the PlatoonSAFE[1] tool, which is an extension of the PLEXE simulator to enable the evaluation of fault tolerance regimes and real-time prediction of communication delays. More specifically, we describe the implementation and usefulness of the module Runtime Manager (RTM) included in PlatoonSAFE that enables monitoring transient communication outages during platooning runtime and switching between different platoon controllers and/or adjustments of inter-vehicle gaps according to some predefined safety rules. The RTM module of PlatoonSAFE models the transient communication outages into finer granularities instead of the traditional way of considering a communication link as either present or absent [7] or assuming a constant communication delay, e.g., [6]. The rationale behind such modeling of wireless connectivity is that it allows us to attribute a platoon performance as a function of the level of communication outage to keep a platoon fail-operational during cruising. Researchers of other domains, e.g., control theory or traffic engineering, can use the RTM module to evaluate platoon safety by considering a more realistic behavior of wireless communications. In contrast to temporary outages or faults, a platoon should perform safe stops to transition to a fail-safe state in case of, e.g., hardware failures, permanent failures, or hazardous road conditions. To this end, PlatoonSAFE includes the implementation of event-driven messages, e.g., Decentralized Environmental Notification Messages (DENMs) [21], with the help of which emergency braking can be simulated. PlatoonSAFE already includes several emergency braking strategies in the emergency braking module to facilitate the evaluation of transitioning a platoon to a fail-safe state. Moreover, in order to put intelligence onboard the platooning vehicles instead of at the network edge or cloud only, we detail the integration and usage of ML models that enable online training and real-time prediction of communication delays incurred by the platooning vehicles. In addition, necessary guidelines are provided so that the ML module can also be used as an online prediction tool to predict different communication, network, and traffic parameters in real-time to facilitate different application-specific QoSs. Rigorous simulation studies have been conducted to demonstrate the impacts of transient communication outages on platoon safety and validate the modules of PlatoonSAFE. The simulation results reveal the necessity of fine-grained modeling of transient communication outages in designing platoon functions and evaluating platoon safety.

The rest of the paper is arranged as follows: Section II first describes the characteristics of some mobility and network simulators that are the necessary basic building blocks of any platooning simulator. Then the platooning simulators currently available to the community are described in Section III. Next, Section IV presents the structure of the PlatoonSAFE tool, its features, and implementation details. In Section V, we present the scenarios that can be simulated using PlatoonSAFE and its configuration parameters and evaluation metrics. In addition, section VI presents the simulation results of some use cases as proof of concept. Finally, Section VII describes potential extensions of PlatoonSAFE, and Section VIII concludes the paper.

1. https://github.com/shahriarHasan09/PlatoonSAFE

II. COMPONENTS OF A PLATOONING SIMULATOR

Jia et al. [22] characterize a platoon as a Vehicular Cyber-Physical System (VCPS) in which vehicle mobility and dynamics represent the physical plane, and the vehicular network represents the cyber plane. The physical and cyber planes are tightly coupled, and one plane can significantly impact the performance of another. For instance, temporary communication outages can cause platoon instability, lead to tracking errors with the LV, or even inter-vehicle collisions. Therefore, a good platooning simulator must consider this interdependency and facilitate bi-directional coupling between network and traffic mobility simulators.

A. TRAFFIC MOBILITY SIMULATORS

A traffic mobility simulator enables the realistic simulation of vehicle movements and their behaviors in the presence of traffic. Traffic mobility simulators for VANETs can facilitate microscopic and/or macroscopic-level simulations. Microscopic-level mobility simulations allow defining individual vehicles' movements so that a user can define, e.g., vehicle types, dynamics, speed, acceleration, car following models, routes, lanes, vehicle positions in the lane, and more. Examples of macro-mobility are importing external maps into simulations, characterizing road topology, defining speed limits, number of lanes, rules at intersections, etc., [9].

SUMO [23] is an open-source and highly portable traffic mobility simulator enabling microscopic-level simulations. It facilitates realistic and large-scale simulations with an unlimited network size and an unlimited number of vehicles. Furthermore, SUMO can be bidirectionally coupled with network simulators to control the mobility of vehicles based on inputs from network simulators. VanetMobiSim [24] also supports the vehicles' macroscopic and microscopic mobility to simulate realistic vehicle movements. It allows for defining topological maps, obstacles, vehicle characteristics at the microscopic and macroscopic levels, path motion, intersection rules, etc. PTV VISSIM [25] is a commercial microscopic traffic simulation software facilitating rich Graphical User Interface (GUI) support with 3D visualization. PTV VISSIM can be linked with external programs through, e.g., Dynamic Link Libraries (DLL). The main advantage of SUMO over PTV VISSIM is that SUMO is open-source and well-maintained. Furthermore, SUMO provides rich GUI support and allows defining the types of vehicles, e.g., cars or trucks, vehicle colors, lengths, dimensions, etc., and considers that different types of vehicles have different speeds and acceleration capabilities. These microscopic-level characteristics are crucial to simulate a platooning scenario with heterogeneous vehicles.

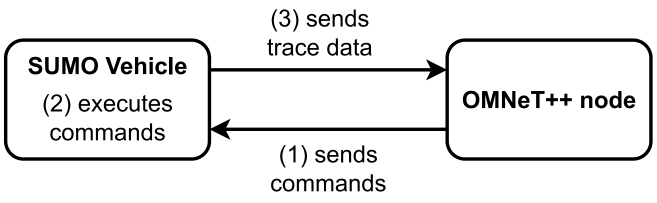


FIGURE 1. Illustration of bidirectional coupling between OMNeT++ and SUMO in Veins simulator.

B. NETWORK SIMULATORS

A network can be modeled by slicing the continuous time and assigning the discrete sequence of network events to the time slices such that two consecutive events can capture the change of network states; such modeling of a system is known as discrete-event simulation. Discrete event simulators such as NS-2,[2] and NS-3,[3] and OMNeT++ [26] have been extensively used for simulating different types of communication networks in the recent past. NS-2 is open source, and it is particularly popular for simulating, e.g., Transmission Control Protocol (TCP) and routing protocols over wired and wireless networks. NS-2 is developed using C++ language, but TCL language is used for network configuration. NS-3 also uses C++ and facilitates the reusing of in-built modules due to its modular architecture. Many NS-3 modules, such as different types of routing protocols, Carrier Sense Multiple Access (CSMA) protocol, Long-Term Evolution (LTE), Wi-Fi, etc., are available to the community. OMNeT++ is an open-source, C++-based, highly modular discrete event simulator. In OMNeT++, simple modules representing, e.g., Network Interface Card (NIC), mobility, protocols, etc., are implemented using C++ classes. The simple modules can be grouped together to form a compound module, e.g., a network node such as a car or User Equipment (UE). The modules in OMNeT++ communicate through messages. In OMNeT++, network description files (.NED) are used to compose compound modules by combining simple modules to set up a network, and initialization files (.ini) are used for parameter definitions. OMNeT++ provides strong GUI support, and OMNeT++-based simulation frameworks are easy to extend due to their modularity characteristics. According to Zarrad and Alsmadi [27], OMNeT++ is excellent in terms of Integrability, Reusability, and Flexibility and has lower complexity than NS-2 and NS-3.

C. BIDIRECTIONAL COUPLING

For realistic simulations of vehicular networks, bidirectional coupling between traffic mobility and network simulators is required. Take the VANETs simulator Veins [19] that bidirectionally couples OMNeT++ and SUMO, which is illustrated by Figure 1. In Veins, a node in OMNeT++ has a corresponding vehicle in SUMO. OMNeT++ sends a series of commands at a particular timestamp to SUMO, and SUMO executes these commands and sends back the vehicle's position so that OMNeT++ can update the position of the corresponding node. This way, OMNeT++ advances the simulation of SUMO at discrete time events through tight coupling. Such bidirectional coupling between OMNeT++ and SUMO in Veins is provided by TraCI [28], a TCP-based open-source client/server interface. Several other VANETs simulators available to the research community also integrate network and traffic simulators through TraCI. For instance, TraNS [29] uses TraCI to couple NS-2 and SUMO to facilitate large-scale VANETs simulations. iTETRIS [30] is another VANETs simulator that couples NS-3 and SUMO. Moreover, iTETRIS uses a custom-made parser named iTETRIS Control System to handle the bidirectional coupling.

III. RELATED WORKS

A plethora of simulation platforms and models have been proposed over the years for the safety evaluation of CACC strings and vehicle platoons. In general, the proposed simulation platforms in the literature can be categorized into open-source and closed-source platforms. The advantages of open-source simulation platforms are that researchers from different domains can contribute together to build a platform encompassing a wide range of features, and the research results are verifiable. To this end, we list some notable features below that a platooning simulator should have to facilitate the evaluation of platoon safety:

- It should be open-source so that the results are reproducible and the community can use it or extend it further.
- It should have a full-fledged network stack with, e.g., physical layer channel models, MAC protocols, and the implementation of different kinds of platoon control messages, e.g., Cooperative Awareness Messages (CAMs) [31], DENMs, etc.
- A generic model for relaying platoon control messages.
- Large-scale simulations under mixed traffic scenarios and the neighboring traffic should be V2V enabled as well.
- Rules for the interaction between CAVs and Human-driven Vehicles (HVs) to enable, e.g., vehicle join, cut-in/ cut-out, and leave maneuvers. Moreover, rules for platoon forming, merge, and split maneuvers should also be present. In addition, such maneuvers should be verifiable under time-varying communication delays.
- Human-machine interaction to simulate, e.g., driver intervention.
- Simulation of urban and highway scenarios.
- Realistic road traffic and road condition simulations.
- Implementation of different types of vehicles, e.g., cars, trucks, lorries, trailers, etc., with realistic vehicle dynamics and engine models.
- Implementation of platoon control algorithms.

2. https://www.isi.edu/nsnam/ns/
3. https://www.nsnam.org/

- Fault tolerance features to evaluate the behaviors of vehicles in the events of transient errors, permanent failures, or hazards.
- Integrated ML models for predicting communication, network, and traffic parameters and enabling a high level of autonomy.
- Teleoperated driving.
- Large-scale and parallel simulation runs and efficient results collection and evaluation methods.

In the rest of this section, we analyze the recent simulation platforms in terms of the features mentioned above.

van Arem et al. developed MIXIC [32], a microscopic traffic simulation model, which can generate road traffic and simulate lane-changing maneuvers and a car following model. Moreover, the authors implement an Autonomous Intelligent Cruise Control (AICC) model that can interact with the onboard driver to perform braking. Lei et al. [33] developed a simulation framework in which CACC is implemented in Simulink, SUMO is used as a traffic simulator, and OMNeT++ is used for network simulation. The authors study the effects of packet losses on a CACC system. Gechter et al. present the VIVUS simulator in [34] that can create a virtual prototype of a vehicle considering its physical properties, sensors required for control algorithms, and 3D rendering. Guériau et al. extended VIVUS to develop the VIPS simulator [35]. In addition to the VIVUS features, VIPS implements the insertion of various noise levels in sensors to study the effects of sensor errors in platooning. However, neither VIVUS nor VIPS contains the implementation of a network stack. Zhao and Sun [36] propose a simulation framework to evaluate ACC and CACC algorithms using the VISSIM traffic simulator. The authors also implement platoon forming, joining, and splitting maneuvers. However, the simulators developed in [33] and [36] are not publicly available to the community.

Choudhury et al. present a platooning simulator in [37] that uses NS-3, MATLAB, and VISSIM for simulating communication networks, control algorithms, and road traffic, respectively. The authors implemented the CACC controller proposed by Rajamani [38] and studied the LV tracking ability of the FVs in a platoon. However, this simulator is not available to the community. Segata et al. developed the platooning simulator PLEXE [20] as an extension of the VANETs simulator Veins. As discussed earlier, Veins provides bidirectional coupling between OMNeT++ and SUMO. In addition, Veins implements the IEEE 802.11p protocol stack [5] and provides several Physical (PHY) layer channel models. As an extension of Veins, PLEXE incorporates platooning maneuvers, e.g., platoon forming and joining to Veins. Moreover, PLEXE implements several controllers, realistic vehicle dynamics, and engine models in the SUMO part. Mena-Oreja and Gozalvez [39] extend the PLEXE simulator to implement platooning maneuvers such as merge, split, and leave. The authors use Python scripts to intercept vehicle parameters from the TraCI interface and facilitate safe platooning maneuvers. Amoozadeh et al. developed the VENTOS simulator [40] that facilitates realistic platooning simulation using the Veins and SUMO simulators. In VENTOS, roadside units, obstacles, and vehicles can be inserted, and the vehicles' speed can vary during the runtime. VENTOS also provides hardware-in-the-loop features to emulate scenarios with real hardware. Malik et al. developed the ComFASE tool [41] as an extension of PLEXE, which facilitates the evaluation of the effects of V2V communication delay attacks and denial-of-service attacks on vehicle platooning. In addition, Vita et al. set up a simulation environment [42] by integrating the OMNet++-based simulator SimuLTE [43] and ML framework Keras to test LTE/5G scenarios. Nonetheless, the ML model is executed by an external python module, and the information exchange is made using text files. In [44], Segata et al. propose that a vehicle should use multiple communication interfaces, i.e., IEEE 802.11p, Visible Light Communication (VLC) and LTE, and failure of a communication interface lead to switching between different control algorithms. The authors provide guidelines on how to link PLEXE with external libraries such as Veins, Veins VLC [45] and SimuLTE.

Hu et al. propose a simulation platform to evaluate CACC in urban scenarios [16]. The platform provides models for simulating different CAV maneuvers and the interaction of the CAVs with HVs under different traffic scenarios in an urban environment. Moreover, the authors propose metrics for quantifying, e.g., traffic capacity, delays, and safety, and validate the simulation platform through comparison with field test results. In [15], Lai et al. propose a generic simulation platform to facilitate the simulation of large-scale road traffic, enable the simulation of multiple CACC controllers, and integrate vehicle dynamics into simulations. Moreover, the authors propose a generic decision-maker to enable cooperative lane-changing maneuvers in CAVs. The simulator is developed using the Application Programming Interface (API) of the commercial traffic simulator VISSIM. In [46], Liu et al. also address the interaction of CACC and HVs and devise rules for automated lane changing and speed control in, e.g., highway entry-exit ramps. Cui et al. propose a simulation platform in [6] to simulate vehicle dynamics, CACC controllers, effects of communication and sensor failures, crash severity, and cyber-attack. However, this simulation platform assumes a constant communication delay for modeling communication behavior, and also it is not an open-source platform. Xiao et al. propose an ACC-CACC car-following model in [47] and simulate a driver intervention model, i.e., a human driver takes over the CACC longitudinal control. The simulations are carried out using the MATLAB simulator. The simulation platforms mentioned above investigate platooning maneuvers in mixed traffic scenarios, with simplified assumptions made about the communication channel. For instance, ideal communication within a specific range (e.g., 200 or 300 meters) or constant communication delay (e.g., 50 ms) is often assumed. However, these works do not consider the transient nature

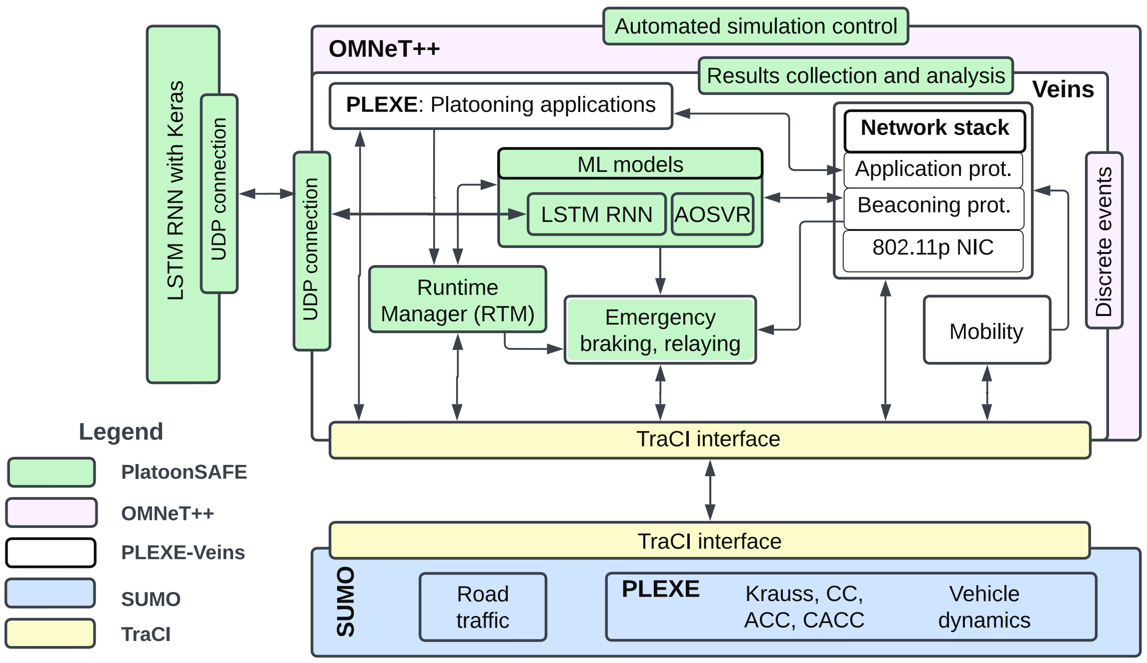


FIGURE 2. Schematic representation of PlatoonSAFE tool.

of wireless connectivity due to path loss, multipath fading, shadowing, attenuation, etc., and unbounded channel access delays that can be caused by neighboring traffic, especially in a dense traffic scenario that is common in an urban environment. PlatoonSAFE enables realistic modeling of transient communication outages that can be used to evaluate, e.g., the platooning maneuvers or rules proposed in the simulation platforms mentioned above and fault tolerance mechanisms. Moreover, PlatoonSAFE is open-source and does not use commercial simulation software, fostering its accessibility and customization potential.

To the best of our knowledge, an open-source platooning simulation framework that facilitates the evaluation of fault tolerance regimes, i.e., fail-operational and fail-safe states in the presence of transient communication outages, and integrates ML models as intrinsic features to predict communication parameters during runtime is nonexistent in the community; this paper aims to close this gap.

IV. STRUCTURE, FEATURES, AND IMPLEMENTATION DETAILS OF PLATOONSAFE

A schematic diagram of the PlatoonSAFE tool is presented in Figure 2. The blocks highlighted in green represent the PlatoonSAFE features, and the rest of the features illustrated in Figure 2 are inherited by PlatoonSAFE from Veins and PLEXE simulators. PlatoonSAFE inherits the implementation of the IEEE 802.11p protocol stack and several realistic PHY layer channel models from Veins. These features of Veins are implemented in the OMNeT++ part as shown in Figure 2. In addition, Veins facilitates microscopic-level simulations of large-scale and mixed traffic scenarios through SUMO. Moreover, PlatoonSAFE inherits the implementation of several state-of-the-art ACC and CACC controllers, realistic engine models, and vehicle dynamics from the PLEXE simulator; these features are implemented in the SUMO part in Figure 2. Furthermore, PLEXE enables the realistic simulation of mixed traffic scenarios, i.e., platoons can have communication-enabled neighboring vehicles that can generate data traffic using periodic messages, causing the platooning vehicles to experience transient communication outages. In addition, platoon forming and join maneuvers and beaconing protocols are implemented in PLEXE. However, what can be simulated with PlatoonSAFE is not limited to the features illustrated in Figure 2 as it builds upon open-source and widely used simulators such as Veins, PLEXE, and SUMO, i.e., other researchers can further extend it. In summary, PlatoonSAFE facilitates the evaluation of its features under realistic considerations of automotive, transportation, and communication features.

PlatoonSAFE mainly comprises three separate modules: the Runtime Manager (RTM) module, the emergency braking module, and the ML module, as shown in Figure 2. The RTM module is responsible for fine-grained modeling of transient communication outages and attributing a platoon performance level according to some predefined safety rules. The emergency braking module implements several braking strategies, and the ML models Long Short-term Memory (LSTM) Recurrent Neural Network (RNN) and Accurate Online Support Vector Regression (AOSVR) are implemented in the ML module. The RTM module, emergency braking module, and the AOSVR model are implemented in the OMNeT++ part of PlatoonSAFE; see Figure 2.

Additionally, the LSTM RNN model is trained online in an external python module with the communication delay values, which are communicated to and from the OMNeT++ part through a UDP socket.

PLEXE gains access to RTM through the `onPlatoonBeacon` method of the RTM module. This method is invoked every time a vehicle receives a periodic beacon, e.g., CAM, from the preceding vehicle or the LV. The braking strategies are implemented using event-driven messages, such as DENMs, which are disseminated when an imaginary road hazard is encountered. The order and rate at which the vehicles perform braking depend on the selected braking strategy. Additionally, the DENMs can be relayed through the middle vehicle in the platoon. The RTM module and the braking strategies can be activated together or separately. If focusing only on the braking strategies, the RTM module can be disabled, and vice versa. The ML models can be activated to predict the communication delays experienced by the last vehicle in the platoon during runtime. The predicted delays can be used to perform, e.g., delay-aware emergency braking maneuvers [48]. Finally, PlatoonSAFE facilitates the automation of simulations, results collection, and analysis through several python scripts. These scripts can also run large-scale and parallel simulations by automatically configuring simulation parameters.

In the documentation, we detail the version control information, downloading and installation guide, line-by-line explanation of the input parameters, detailed explanation of the implementations, and the guidelines for using the PlatoonSAFE features. However, to explain the PlatoonSAFE features and simulation scenarios in this paper, we refer to some files in the source code; we refer the readers to the documentation where direct links to the source code of these files are provided.

A. CONTROL ALGORITHMS IMPLEMENTED IN PLEXE

PLEXE, the predecessor of PlatoonSAFE, implements several controllers in the SUMO part, as illustrated in Figure 2. A *controller* regulates the speed, acceleration, and inter-vehicle gaps in a platoon based on the data obtained through onboard sensors and wireless communications. In this section, we do not describe all the controllers implemented in PLEXE; instead, the controllers used for developing the RTM module of PlatoonSAFE are described. In ACC [49], a vehicle uses its onboard sensors, e.g., radar and/or laser, to detect the speed changes of the vehicle in front and maintain the desired gap set by the driver. If no vehicle is in front, the vehicle drives at the desired speed like a traditional Cruise Control (CC). CACC adds V2V communications to ACC in order to facilitate short inter-vehicle gaps and string stability. A platoon is said to be string stable when the disturbances such as speed and distance errors are attenuated towards the tail of the platoon [20]. Ploeg et al. propose a CACC controller [50] in which a platooning vehicle computes its acceleration based on the distance measured through its radar

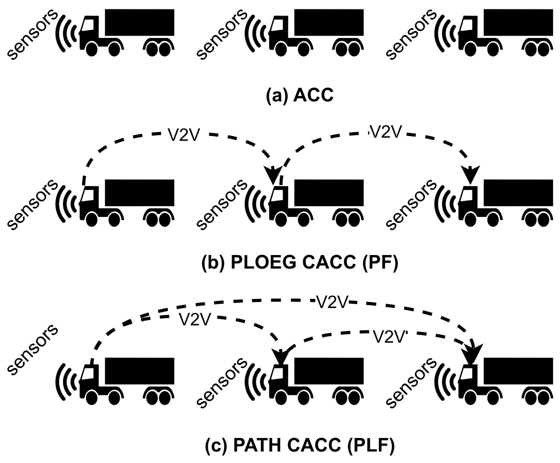


FIGURE 3. Sensing and communication topologies of different controllers.

sensors and kinematic parameters of the preceding vehicle obtained through V2V communications. As a vehicle using PLOEG CACC receives information from its preceding vehicle only, it is also regarded as Predecessor Following (PF) strategy as illustrated in Figure 3. ACC and PLOEG CACC rely on the Constant Time Gap (CTG) policy, i.e., the inter-vehicle gaps in a platoon using the ACC or PLOEG CACC controller vary with the change of speed. On the other hand, the PATH CACC controller [51] proposed by Rajamani et al. uses the Constant Distance Gap (CDG) policy in which the desired gap is always maintained despite speed changes. Moreover, in the PATH CACC controller, the desired gap to the preceding vehicle is computed based on periodic beacons received from the vehicle in front and the LV through V2V communications. To this end, the communication topology of PATH CACC is regarded as Predecessor-Leader Following (PLF) strategy as depicted in Figure 3. Segata shows that a platoon using ACC requires a 1.2 s time gap (35.35 m at 100 kmh^{-1}) to be string stable [52], whereas the PLOEG CACC exhibits string stability with 0.5 s time gap (15.89 m at 100 kmh^{-1}) [50]. However, PATH CACC allows inter-vehicle gaps as short as 5 meters due to the inclusion of V2V communications with the LV and adopting the CDG policy [52].

B. IMPLEMENTATION OF RUNTIME MANAGER (RTM) FACILITATING THE EVALUATION OF FAIL-OPERATIONAL PLATOONING DURING TRANSIENT COMMUNICATION OUTAGES

In this subsection, we first explain the concept of *Contracts* that are used for assigning a platoon performance level as a function of the duration of communication outages in the RTM module. Then the control flow of RTM and the implementation details of the contracts are presented.

1) DEGRADATION CASCADE EMPLOYING CONTRACTS

The idea of keeping a platoon fail-operational during transient communication outages is that a platooning vehicle

TABLE 1. Example of strong and weak contracts.

A	deceleration$_{max}$ = -8 ms^{-2} AND deceleration$_{min}$ = -1 ms^{-2} AND PATH CACC, PLOEG CACC, or ACC is active;
G	Platooning vehicles avoid collisions during cruising and emergency braking AND a hazard is circumvented at a full stop.
B$_1$	PATH CACC active AND no local control failure AND no radar/lidar failure AND *good* communication quality with the LV;
H$_1$	PATH CACC can be retained with a 5 m gap AND a high fuel efficiency is attained;
B$_2$	PATH CACC active AND no local control failure AND no radar/lidar failure AND DENM received from the LV instructing to brake at -8 ms^{-2} within 10 ms of encountering a hazard;
H$_2$	Emergency braking performed AND the vehicle transitions to a fail-safe state;

degrades its performance by increasing the gap to its front vehicle and/or adopting a less fuel-efficient controller proportionally with the duration of communication outages [53]. When various levels of communication outages dictate the formation of an ordered set of platooning operation modes, it is regarded as a *degradation cascade* [54]. The behavior of a system, e.g., a platoon, in various degraded operational modes can be captured by a set of Assumption/Guarantee contracts [8]. A contract $C =< A, G >$ can be defined as a pair of properties in which A represents the *Assumptions* on the system environment and G represents the *Guarantee* that the system promises, given that the assumptions are fulfilled [55]. Sljivo et al. distinguish between strong contract C and weak contract $C_{weak} =< B, H >$, where B and H are weak Assumption and weak Guarantee, respectively [56]. Take Table 1 as an example that is adopted from [2] and [8]. The strong Assumption A and Guarantee G in Table 1 represent the overall safety goal of a platoon such that A must always hold in all conditions and G is always fulfilled. In contrast, the weak guarantees, e.g., H_1 and H_2 in Table 1, only require holding when weak assumptions B_1 and B_2 are satisfied in addition to the strong Assumption A. Unlike strong assumptions, weak assumptions may not always hold. For instance, the weak Assumption B_2 in Table 1 represents that a DENM is received from the LV within 10 ms of encountering a road hazard, which might not be satisfied due to transient communication outages. In such a case, the weak Guarantee B_2 may not hold, i.e., the vehicle may not transition to a fail-safe state. Therefore, a weak guarantee holds only when both the strong and weak assumptions are fulfilled in the system environment.

Performance degradation as a way of keeping a platoon acceptably safe during cruising has been studied in the literature. For instance, Yu et al. [57] propose to degrade platoon performance to attain string stability in the events of communication interruption. Ploeg et al. propose to adopt a "degraded CACC" mode during time-varying communication delays [58]. Sljivo et al. propose a set of safety contracts to keep a platoon acceptably safe in the presence of failures [8]. Girs et al. propose a contract-based runtime monitoring system to assure the safety of vehicular systems in the presence of wireless communication outages [59]. In previous work by the authors [2], rigorous simulation studies demonstrate that degrading and upgrading platoon performance as a function of the level of communication outages following safety contracts can keep a platoon acceptably safe. Although the concept of safety contracts for assuring degradation cascades in vehicle platooning has been studied rigorously in the literature, the existing simulation platforms do not facilitate the simulation of contracts under realistic vehicle dynamics, road and data traffic, and communication network scenarios. The RTM module presented in this section aims to address this gap.

The fail-operational method implemented in PlatoonSAFE called the Runtime Manager (RTM) is based on a set of Assumption/Guarantee contracts, which define the inter-vehicle gap and the controller a vehicle has to adopt (Guarantee H), given its experienced communication quality with the vehicle in front and the LV (Assumption B). Although we consider only transient communication outages as a basis for defining the contracts, PlatoonSAFE can be further extended to define contracts that include all kinds of failures (temporary and permanent) in the automotive domain. For example, the weak guarantees in Table 1 can be adopted to define what happens if there is a failure in the local control system or distance sensors. Moreover, the guarantees may also reflect what measures should be taken in case of, e.g., slippery road conditions, bad visibility, warning from tire pressure sensors, cut-in/cut-out scenarios, platoon join, split, and merge maneuvers, etc. To this end, we first describe the contracts implemented in the RTM module that considers different levels of communication outages to define the Assumption/Guarantee contracts, called the *default contracts*. Next, a general framework is described in which contracts can be defined in a high-level language. As PlatoonSAFE is developed on top of open-source simulators, researchers from different domains can adopt the contract list to study different kinds of failures in the automotive domain.

2) RTM CONTROL FLOW

The contracts in the RTM module are implemented in the OMNeT++ part of Figure 2, and the TraCI interface is used to send commands to the SUMO vehicles. In the RTM module, we use the OMNeT++ `handleSelfMessage` function in the `RuntimeManager.cc` file, which is recursively called at an interlude of `rmMonitorInterval` using the `ScheduleAt` function of OMNeT++, see Algorithm 1. RTM performs several tasks by periodically calling the `handleSelfMessage` method until the simulation time limit is reached. First, RTM *logs* the ego vehicle data, such as the currently active controller, maximum deceleration rate, current gap to the vehicle in front, and simulation time. If the gap to the front vehicle is smaller than a predefined safety gap, a *safety violation report* is recorded in an output file. The main task of RTM, i.e., switching between different

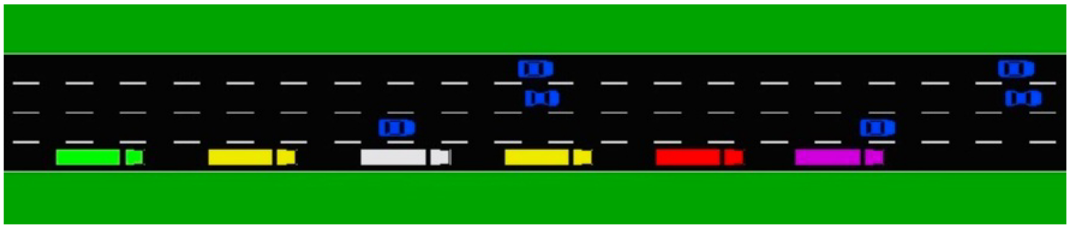


FIGURE 4. A screenshot of SUMO GUI demonstrating a platoon on the rightmost lane and the neighboring vehicles on the left lanes.

Algorithm 1 `handleSelfMessage ()`

INPUT: cMessage *message

```
1: if $message = monitoringMsg$ then
2:     callBackTime = simTime() + rmMonitorInterval;
3:     egoLog();
4:     safetyViolationCheck();
5:     evaluate();
6:     scheduleAt(callBackTime, monitoringMsg);
7: end if
```

TABLE 2. Example default assumption/guarantee contracts.

Action Type	Assumption			Guarantee	
	c2f	*c2l*	active controller	active controller	gapToFront
Degradation	*good*	*good*	PATH CACC	PATH CACC	DEFAULT
	fair	*good*	PATH CACC	PATH CACC	INCREASE
	poor	*good*	PATH CACC	PLOEG CACC	DEFAULT
	fair	*fair*	PATH CACC	PLOEG CACC	INCREASE
	fair	*poor*	PLOEG CACC	PLOEG CACC	INCREASE
	poor	*poor*	PLOEG CACC	ACC	DEFAULT

states based on the experienced communication quality, is carried out by the `evaluate` method, which is implemented in the `Contracts.cc` file. RTM searches for a contract from the default contract list for the current *Assumptions*, i.e., experienced communication quality and active controller in the ego vehicle. If a contract is found in the contract list, the corresponding *Guarantee* is provided.

3) IMPLEMENTATION OF ASSUMPTION/GUARANTEE CONTRACTS

A platooning vehicle using the RTM monitors the *connection to the front (c2f)* vehicle and *connection to the lead (c2l)* vehicle at an interval of `rmMonitorInterval`. Instead of the traditional way of treating communication as either present or absent, the *c2f* and *c2l* are classified into *good*, *fair*, and *poor* communication qualities that are defined by the *duration of communication outages* experienced by a vehicle with respect to its front and LV. The duration of the communication outage is calculated as the simulation time difference between the currently received CAM and the last CAM received by the ego vehicle. Table 2 depicts a subset of the Assumption/Guarantee contracts implemented in RTM as an example. Please refer to the documentation and source code for the full list of default contracts. From Table 2, we can see that the ego vehicle's *c2f*, *c2l*, and currently active controller are the *Assumptions* in the platooning environment. There can be four types of *Guarantees*: 1) *transitioning* to another controller; 2) *increasing* the gap to the vehicle in front; 3) both controller switching and increasing the gap; 4) retaining the current controller and the *default gap*. As an example, we can derive a weak contract C_{weak} from Table 2 written in a semi-formal language as: <B = PATH CACC active AND *c2l* is *good* AND *c2f* is *poor*; H = transition to PLOEG CACC>.

As different platooning vehicles are likely to experience different levels of communication outages, each vehicle uses the RTM independently. Therefore, vehicles in a platoon may have heterogeneous controllers based on the *c2f* and *c2l*. A platooning vehicle adopts the PATH CACC when *c2f* and *c2l* are *good*. The rationale is that a vehicle using PATH CACC requires feedback information both from the LV and the preceding vehicle to maintain short gaps and, consequently, high fuel efficiency. However, if *c2f* becomes *fair*, a vehicle retains PATH CACC but increases the gap to the vehicle in front. *Poor* communication quality is considered a temporary communication outage; therefore, a vehicle adopts PLOEG CACC due to *poor c2l*, as PLOEG CACC does not require feedback information from the LV. If *c2f* becomes *fair*, PLOEG CACC is still retained but with an increased gap. The adoption of PLOEG CACC is considered a performance degradation in terms of fuel efficiency because a longer gap than PATH CACC is required here. Finally, *poor c2f* leads to adopting the ACC controller, which does not rely on V2V communications. Notice that Table 2 forms a series of degradation cascades, and the contracts can represent different levels of the cascades. Moreover, a vehicle first increases the gap to the vehicle in front due to *fair* quality instead of directly switching to another controller to facilitate *graceful performance degradation*. Like degradation, *performance upgradation* is performed by returning to a more fuel-efficient controller, e.g., PATH CACC, and adopting the default gap when communication quality improves. The upgradation cascades are also defined by Assumption/Guarantee contracts; please refer to the source code for the full list.

The SUMO part of PlatoonSAFE provides strong GUI support to facilitate the visualization of the platooning states adopted by the vehicles; see Figure 4. Several colors represent the states of the platooning vehicles on the rightmost lane. For instance, purple, red, and green represent ACC, PATH CACC, and PLOEG CACC controllers, respectively.

Furthermore, yellow and white colors represent PATH CACC and PLOEG CACC with increased gaps.

4) USER-DEFINED CONTRACTS

The RTM module in PlatoonSAFE is developed considering the necessity of evaluating platoon safety for different kinds of transient/permanent failures, e.g., communications, sensors, and actuator failures. To that end, PlatoonSAFE users can define contracts in the `contracts.txt` file of the RTM module using a high-level language and following certain input formats. A parser converts the inputs to an object of C++ `Contract` class type in the RTM module. The default safety contracts are implemented using the `map` container of the C++ Standard Template Library (STL). Moreover, C++ polymorphism has been extensively used in developing RTM to induce further extension. Please refer to the documentation for further details on formatting user-defined contracts.

C. IMPLEMENTATION OF ML MODELS FOR PREDICTING COMMUNICATION DELAYS

The ML module in PlatoonSAFE aims at facilitating online and real-time prediction of communication, network, and traffic parameters. The use of ML for making such predictions has received significant research attention in the recent past. Several works suggest collecting data through simulations and training the ML models offline to make predictions, e.g., [14], [60]. However, as the network and communication quality between CAVs change frequently, online predictions are of the essence here. Jornod et al. introduce a multi-layer perceptron regressor model that can forecast Packet Inter-Reception time (PIR) in vehicular networks [3]. The authors not only assess the accuracy of the model but also investigate its capability for online training and runtime prediction of PIR. Moreover, Torres-Figueroa et al. report that the prediction of end-to-end delays in a cellular network without the information from the base station leads to low prediction accuracy, which is inadequate for making safety-critical decisions [12]. Therefore, it is worth investigating the possibility of online training and predicting communication and network parameters in self-organized networks, e.g., VANETs. Despite this, the simulation platforms available in the literature do not avail the feature of online training of ML models. To this end, this section describes the integration of LSTM RNN and AOSVR models that constitute the ML module and the prediction of communication delays as an example. Note that the use of the ML module of PlatoonSAFE is not limited to merely predicting communication delays; instead, the models can be used for predicting, e.g., data and packet error rates [13], platooning maneuvers such as cut-in [61], wireless channel idle activities [62], traffic patterns [63], drag force [64], network traffic classification and network flow prediction [65], and more. In the PlatoonSAFE documentation, we provide the details of the integration and implementation procedure that can be followed to use the models for different purposes other than predicting communication delays.

In the example use case of PlatoonSAFE, the ML models are used for predicting the communication delays experienced by the last vehicle in a platoon with respect to its LV because the last vehicle usually experiences the highest delay due to path loss and fading effects. Every time the last vehicle receives a CAM from the LV, it computes the delay by taking the difference between the current simulation time and the time in which the last CAM was received. This delay value is then broadcasted through a `DelayMessage`. When the LV receives the `DelayMessage`, it uses either Long Short-term Memory (LSTM) Recurrent Neural Network (RNN) or Accurate Online Support Vector Regression (AOSVR) to predict the maximum delay in the platoon. The predicted delay is then encapsulated in the upcoming periodic beacon, i.e., CAM, so that all vehicles know the expected maximum delay in the next beacon interval.

Both AOSVR and LSTM RNN models must be trained online, i.e., during platooning runtime, as the communication delays are time-varying due to packet losses, channel access delays, packet collisions, etc. The AOSVR model facilitates the training of new samples (delay values) added to a dataset without having to restart or reprogram the training process of the entire dataset like conventional SVR. The AOSVR that was studied and implemented by Parrella in [66] first evaluates the relevance of all the samples for every new sample added to the training dataset, gives more importance to the recent ones, and forgets the samples that are deemed not to add any new information, and finally, the AOSVR is retrained for the new sample. The C++ version of the AOSVR[4] implemented by Parrella is integrated as an ML module directly into PlatoonSAFE. Three estimation parameters ϵ, C_{SVR}, and *SizeLimit* are required to be optimized for prediction with AOSVR. ϵ defines an error margin in which predictions are not penalized, C_{SVR} is a regularisation parameter that determines the amount of misclassification that shall be avoided, and *SizeLimit* is the number of valuable samples that are taken into consideration for making a new prediction.

In addition to AOSVR, we have decided to use LSTM RNN [67] for online prediction of communication delays because it is widely used for time series predictions [68]. LSTM uses long-term dependencies of data sequence for training and predicting sequential data. Although there are some specific implementations for LSTM or complete libraries in C++, such as *Tensorflow Lite for Microcontrollers*, they do not offer versatility while designing networks or training them during the simulation runtime. To this end, we have implemented an LSTM network in an external python module using Keras[5] on top of Tensorflow,[6] which is an Application Programming Interface (API) that

4. https://github.com/fp2556/onlinesvr
5. https://keras.io/
6. https://www.tensorflow.org/about

simplifies the design and training process for different types of NNs. In order to integrate it into PlatoonSAFE, we have established a User Datagram Protocol (UDP) socket from the simulator to the external python module to exchange information. Whenever the ML module in PlatoonSAFE receives a new delay value, it sends it to the python module through the UDP socket. The LSTM RNN then predicts the next communication delay, which is communicated back to the ML module in PlatoonSAFE, and finally, the weights of the RNN are updated with the new sample. Regarding the structure, we use a simple RNN with one LSTM layer of 32 units and a Dense layer. This allows us to obtain predictions and communicate them to the simulator via UDP faster, which is essential for not slowing down the simulations in PlatoonSAFE. In addition, an Adam optimization model [69] is used for training, with an initial learning rate of 0.001.

D. IMPLEMENTATION OF EMERGENCY BRAKING STRATEGIES

In this subsection, we first describe the control flow of the emergency braking module, which can be used as a baseline for implementing and simulating emergency braking strategies. Moreover, we implement the event-driven messages DENMs and Acknowledgment (ACK) in the emergency braking module that can be used to design braking strategies based on the information available from the LV, preceding vehicle, and/or the following vehicle. In addition, we implement a simple relaying strategy in which the middle vehicle in a platoon can relay the DENMs broadcasted by the LV. This relaying model can be used to implement and evaluate more accurate and complex relaying mechanisms. The means for coordinating between vehicles through wireless communications are implemented in the OMNeT++ part of PlatoonSAFE. However, to perform changes in the dynamics of the vehicles or engine model, modifications are required in the SUMO part; please refer to PLEXE documentation[7] for further details.

PlatoonSAFE also comes with the implementation of several emergency braking strategies, which are described below. In addition, the metrics for evaluating emergency braking strategies in terms of fail-safe conditions are also implemented in PlatoonSAFE.

1) CONTROL FLOW OF EMERGENCY BRAKING MODULE

In order to implement the braking strategies, we use event-driven messages (DENMs). The LV schedules a braking event at a user-defined braking time during the simulation initialization. The `handleSelfMessage` function in Algorithm 2 is called at the user-defined braking time, and DENMs are broadcasted at an interval of `DENMInterval`. In addition, DENMs can be relayed using the middle vehicle in the platoon, which is implemented using the `RelayDENMs` message. When a vehicle either receives a DENM or a relayed DENM, it follows the emergency braking algorithm set by the user.

7. http://plexe.car2x.org/documentation/

Algorithm 2 `handleSelfMessage ()`

INPUT: cMessage *message

```
1: if $message = brakingMessage$ then
2:     sendBrakingMessage(-1);
3:     scheduleAt(simTime() + DENMInterval, brakingMessage);
4: end if
```

2) IMPLEMENTED BRAKING STRATEGIES

The emergency braking module of PlatoonSAFE comes with the following braking strategies by default:

Normal Braking (NB): The NB strategy is a straightforward braking approach in which a vehicle initiates braking immediately upon receiving a DENM from the LV. In case of communication outages, braking is performed by using distance sensors to detect the speed variations of the vehicle in front.

Synchronized Braking (SB): Liu et al. in [70] propose to delay the actuation of platooning vehicles for a short period to cancel out the effects of communication delays on platoon stability. In [48], we propose synchronizing the braking actions of platooning vehicles by delaying the braking until the τ_{wait} period. The τ_{wait} is obtained by taking the average communication delay experienced by the last vehicle in the platoon. Furthermore, the ML models implemented in PlatoonSAFE that predict the communication delays experienced by the last vehicle can be used to predict τ_{wait} in SB.

Gradual Deceleration (GD): In an experimental study, Zheng et al. [71] arrange the platooning vehicles so that the platoon LV brakes at the minimum deceleration rate and the rate increases in the downstream direction gradually such that the last vehicle can brake at the maximum deceleration rate. Using the GD strategy, a vehicle starts braking as soon as a DENM or a relayed DENM is received.

Coordinated Emergency Brake Protocol (CEBP): Bergenhem et al. [72] propose CEBP with the goal of avoiding collisions between the platooning vehicles. In CEBP, the last vehicle brakes first and initiates an Acknowledgement (ACK) packet broadcasting upon receiving a DENM. When the second to last vehicle receives an ACK from the last vehicle, it also starts braking and broadcasts ACK. This way, every vehicle from the tail to the head of the platoon starts braking upon receiving an ACK from its immediate successor.

Adaptive Emergency Braking (AEB): In [10], we propose AEB as an improvement to the CEBP. In AEB, platooning vehicles perform soft braking, i.e., braking at a low deceleration rate, upon receiving a DENM. The soft braking continues until an ACK is received from the immediate successor when a vehicle performs full deceleration. The AEB strategy inherits the collision avoidance property of the CEBP strategy. Furthermore, it minimizes the stopping distance by performing soft braking while waiting for an ACK.

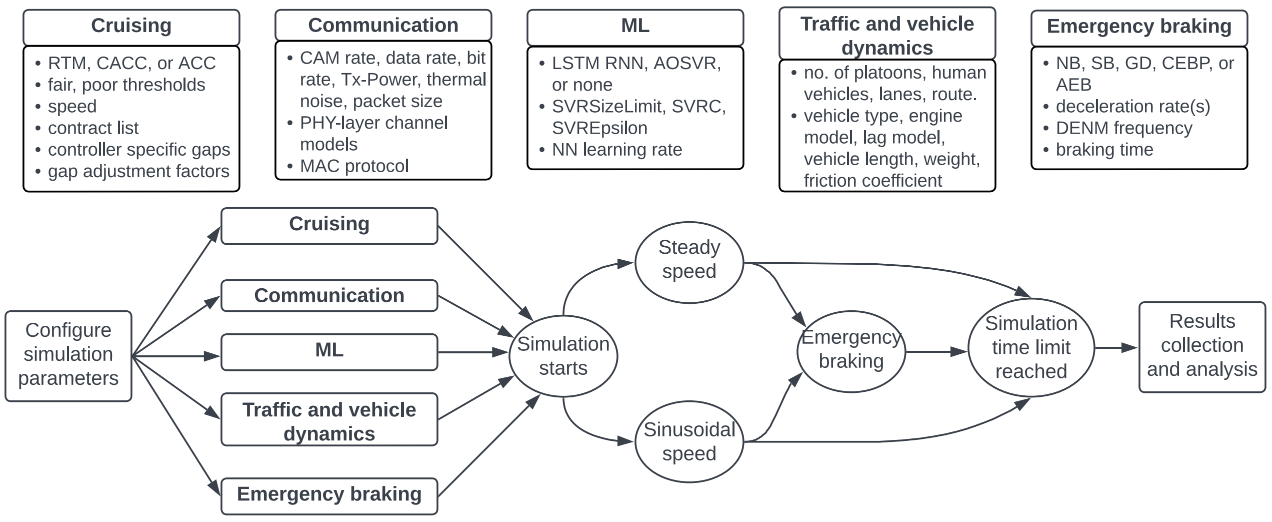

Cruising
• RTM, CACC, or ACC
• fair, poor thresholds
• speed
• contract list
• controller specific gaps
• gap adjustment factors
Communication
• CAM rate, data rate, bit rate, Tx-Power, thermal noise, packet size
• PHY-layer channel models
• MAC protocol
ML
• LSTM RNN, AOSVR, or none
• SVRSizeLimit, SVRC, SVREpsilon
• NN learning rate
Traffic and vehicle dynamics
• no. of platoons, human vehicles, lanes, route.
• vehicle type, engine model, lag model, vehicle length, weight, friction coefficient
Emergency braking
• NB, SB, GD, CEBP, or AEB
• deceleration rate(s)
• DENM frequency
• braking time
Configure simulation parameters
Cruising
Communication
ML
Traffic and vehicle dynamics
Emergency braking
Simulation starts
Steady speed
Sinusoidal speed
Emergency braking
Simulation time limit reached
Results collection and analysis

FIGURE 5. Possible simulation scenarios and configuration parameters with different PlatoonSAFE features.

V. SIMULATION SCENARIOS, EVALUATION METRICS, AND CONFIGURATION PARAMETERS OF PLATOONSAFE

This section describes possible simulation scenarios with PlatoonSAFE, their configuration parameters, and the metrics that can be evaluated.

A. SIMULATION SCENARIOS AND CONFIGURATION PARAMETERS

PlatoonSAFE facilitates the simulation of various combinations of scenarios as depicted in Figure 5. In addition, some notable parameters that are required to be configured to enable the simulation of platoon cruising, communications, predictions, traffic and vehicle dynamics, and emergency braking are listed in Figure 5. Note that the configuration parameters are not limited to the list in Figure 5. During the simulation configuration, the RTM module can be enabled or disabled; if it is disabled, one of the platoon controllers implemented in PLEXE should be activated. In both cases, the inter-vehicle gaps and controller parameters, which are listed in Table 4, need to be set. Running simulations with RTM requires activating either the default contract list or a user-defined contract list. Additionally, the fair and poor thresholds and the factors by which the gaps between vehicles are adjusted in the case of transient communication outages need to be defined. The communication parameters include beaconing intervals of platooning and human vehicles, antenna parameters, the IEEE 802.11p MAC protocol [5], channel models implemented in veins, such as Nakagami-m fading [73], free space path loss model, two-ray interference model [74], and obstacle shadowing model [75]; see Figure 5 and Table 4. Moreover, activation of the LSTM RNN or AOSVR models, along with their parameters, is required for making predictions using the ML module. PlatoonSAFE inherits the traffic and vehicle dynamics models implemented in PLEXE, a subset of which is delineated in Figure 5. Finally, a user must define the emergency braking strategy to be simulated, deceleration rates, DENM intervals, and braking time as listed in Figure 5 and Table 4. We refer the readers to the PlatoonSAFE documentation[8] that includes line-by-line explanations of the configuration parameters and necessary guidelines for using them.

During platoon cruising, we can choose between *sinusoidal* and *steady* speed patterns of the LV. The FVs follow the speed of the LV with the aid of periodic CAMs and/or onboard sensors. In addition, there are options for several emergency braking strategies and two ML models. The cruising pattern, emergency braking strategy, and ML models can generate many simulation combinations. PlatoonSAFE has several such scenarios by default, which can be used as guidelines for generating new scenarios.

B. AUTOMATION OF SIMULATIONS AND EVALUATION PARAMETERS

PlatoonSAFE includes additional python scripts to automate simulations by facilitating the definition of different simulation parameter values, e.g., CAM rates, number of vehicles and lanes, controller, inter-vehicle gaps, etc., which are automatically changed in the simulation settings and all possible combinations of the simulation runs are then executed from the command line. The aim is to run many simulations with different configurations without having to run them one at a time. Simulation results are saved in OMNeT++ vector files (`.vec`), containing statistics on, e.g., speed, inter-vehicle distances, experienced communication delays, prediction errors, etc. Table 3 lists the evaluation parameters that can be used to evaluate different networking and mobility parameters with PlatoonSAFE. Note that PlatoonSAFE inherits the evaluation parameters available in PLEXE and Veins as well.

8. https://github.com/shahriarHasan09/PlatoonSAFE

TABLE 3. Evaluation parameters available in PlatoonSAFE.

Layers	Evaluation parameters
Mobility	acceleration, activeController, controllerAcceleration, distance, posx, posy, relativeSpeed, speed, co2emission, maxSpeed, MinSpeed, totalCO2Emission, totalDistance.
Network & Transport	droppedExceededAttempts, generatedBSMs, generatedWSAs, generatedWSMs, receivedBSMs, receivedWSAs, receivedWSMs, transmissionAttempts, frontDelay, frontTransmissionN, leaderDelay, leaderTransmissionN, predictedDelayAtLV, PredictionErrorAtLV, firstDENMDelay, firstAckDelay,
PHY & MAC	busyTime, totalBusyTime, collisions, droppedPacketsInMac, NumInternalContention, ReceivedBroadcasts, SentPackets ReceivedUnicastPackets, RXTXLostPackets, TotalLostPackets SentAcknowledgements, SlotsBackoff, SNIRLostPackets, TimesIntoBackoff,

PlatoonSAFE also includes a python script that converts the results of vector files into `.csv` files with the following structure: *ParameterName*, *VehicleID*, *SimulationTime*, *ParameterValue*. During this format conversion, the script also computes the following metrics required for analyzing emergency braking strategies:

- *Minimum gap at a full stop:* After a platoon completely stops, the minimum gap between any pair of vehicles in a platoon is computed. A collision is said to have happened if the minimum gap is less than or equal to zero [76].
- *Time to Collision (TTC):* The elapsed time between when the LV encounters a hazard and when the inter-vehicle distance between two vehicles becomes zero or less due to a collision. In case there are collisions between more than a pair of vehicles, i.e., chain collisions, the first collision is considered to calculate TTC. If there is no collision, an empty array is returned by the script.
- *Stopping distance of the LV:* The distance traveled by the lead vehicle from the moment it recognizes a hazard until it reaches a complete stop.
- Time to stop the platoon: The elapsed time between the moment the LV recognizes a hazard and the whole platoon fully stops. This metric can be used to understand the time required by a platoon to transition to a fail-safe state from the moment of encountering a hazard.

The performance of a platoon during cruising using a controller or the RTM module can be evaluated in terms of the ability of the FVs to track the LV, string stability, fuel efficiency, road throughput enhancement, collisions, and more. The simulations with PlatoonSAFE generate data such as speed, inter-vehicle gaps, and relative speed that can be used to evaluate platoon performance during cruising as follows:

- *Safety during cruising:* The RTM module periodically checks for safety violations, i.e., if the inter-vehicle gap is smaller than a user-defined safe inter-vehicle distance threshold, and the safety violation reports are recorded in an output file. In addition, the inter-vehicle gap profiles of the vehicles generated in the simulator can be used to analyze such collisions.
- *LV tracking ability:* How well an FV can track the speed changes of the LV can be analyzed using the speed profiles; examples are given in Section VI.
- *String stability:* The ability to attenuate the disturbances from the head to the tail in a platoon is defined as string stability [38]. If the speed profile of an FV is amplified compared to the speed profile of the LV, it can be regarded as string instability. Moreover, when the disturbances are amplified in the downstream direction, the inter-vehicle gaps also fluctuate, which can be used to recognize string stability as well.
- *Road efficiency:* In this paper, we analyze the distance profiles of vehicles to evaluate road efficiency under the assumption that shorter gaps enable higher road efficiency [15], [18]. However, the road traffic parameters that can be configured during simulations with PlatoonSAFE, such as no. of platoons and vehicles per platoon, no. of platoon lanes, no. of human vehicles, no. of human vehicle lanes, and inter-vehicle gaps, can be used to quantify road capacity. We refer readers to [15] for quantitative evaluation of road capacity.
- *Fuel efficiency:* In Heavy-duty Vehicles (HDVs), the aerodynamic drag of the FVs in platoon reduces drastically with the reduction of inter-vehicle gaps, which leads to a considerable saving in fuel consumption [77]. For quantitative evaluation of fuel efficiency, the vehicle dynamics in the SUMO part of PlatoonSAFE can be configured, and the distance profiles recorded during simulations can be used to calculate the fuel consumption, as shown by the authors in [77] (PLEXE is used in this work). Moreover, Jornod et al. in [78] study fuel consumption reduction in platoons using SUMO. PlatoonSAFE as a successor of Veins also inherits models for calculating CO_2; results of the evaluation parameters such as *co2emission*, *totalCO2Emission* are recorded at the end of each simulation.

In order to evaluate the predicted communication delays using the ML module, the following metrics are used:

- *Root Mean Square Error* $RMSE = \sqrt{\sum_{i=1}^{N}(\frac{(x_i - \hat{x}_i)^2}{N}}$, where x_i is the actual delay and $\hat{x}_i$ is the delay predicted using either AOSVR or LSTM RNN.
- *Predicted delay at LV:* The delay predicted using an ML model at the LV.
- *Prediction error at LV:* The difference between the predicted delay and the actual delay in seconds.

PlatoonSAFE also facilitates the evaluation of different communication parameters as listed in Table 3. The parameters related to CAM and DENM delays are as follows:

- *Front delay:* The time elapsed between the receipt of the current CAM and the previous CAM from the preceding vehicle.
- *Leader delay:* The time duration between receipt of the current CAM and the previous CAM from the leading vehicle.

TABLE 4. Configuration parameters for simulations.

	Parameter	Value
communication	PHY/MAC model	IEEE 802.11p/IEEE 1609.4
	Path loss model	Free space ($\alpha = 2$)
	Fading model	Nakagami-m (m = 1.86)
	Tx Power	100 mW
	Packet size	200 B
	Bit rate	6 $Mbps$
	Sensitivity	-94 dBm
	Thermal noise	-95 dBm
	Frequency	5.89 GHz
	Bit rate (non-platooning vehicles)	3 $Mbps$
mobility	Leader speed	100 kmh^{-1}
	Platoon size	7
	Neighboring vehicles	300 or 400
	No. of platoons	1
	Leader oscillation frequency	0.2 Hz
	Oscillation amplitude	10 $kmph$
	Total no. of lanes	3 or 4
	Platooning vehicles insert time	1 s
	Neighboring vehicles insert time	50 s
	Simulation time limit	$100 \sim 110$ s
controller	Controllers	CACC (PATH, PLOEG), ACC
	Weighting factor C_1	0.5
	Controller bandwidth ω_n	0.2 Hz
	Damping ratio ξ	1
	Controller gain k_p	0.2
	Controller gain k_d	0.7
	PATH CACC CDG	5, 10 m
	PLOEG CACC CTG	0.25, 0.5, 1.0 s
	ACC CTG	0.5, 0.8, 1.2 s
RTM	rmEnabled	$true$
	rmMonitorInterval	0.1 s
	constantSpacingFactor	0.25
	timeGapFactor	0.25
	poor	800 ms
	fair	100 ms
Braking	CEBEnabled	$true$
	brakeAtTime	100 s
	softDecelerationRate	-3 ms^{-2}
	fullDecelerationRate	-8 ms^{-2}
ML	SVR ϵ	0.0001
	C_{SVR}	0.03
	SVR Size Limit	5
	NN Learning rate	0.001

- *First DENM delay:* The elapsed time between generating the first DENM and the time when a vehicle receives a DENM.

VI. PROOF OF CONCEPT

This section evaluates some simulation scenarios related to platoon cruising, emergency braking, and predictions using the evaluation metrics described in the previous section. The aim is to understand the impacts of temporary communication outages on platoon safety.

A. SIMULATION SETTINGS AND SCENARIOS

We simulate a platoon of seven vehicles cruising at 100 kmh^{-1} in the rightmost lane. The configuration parameters and their corresponding values are listed in Table 4. In order to generate mixed-traffic scenarios and realistically capture the effects of data and road traffics on wireless communication outages, we consider the configurations in Table 5:

TABLE 5. Configurations for dense and moderate data and road traffic.

traffic type	Neighbouring traffic				Platoon beacon frequency (Hz)		
	vehicles	vehicles/km	beacon frequency (Hz)	packets s^{-1}km^{-1}	CAM	DENM	ACK
Config 1	400	95	50	4750	10	10	10
Config 2	300	65	20	1300	15	15	15

- *Config 1:* In this configuration, a dense traffic scenario is represented with 400 neighboring vehicles broadcasting beacons at a frequency of 50 Hz. The platooning vehicles transmit CAMs, DENMs, and ACKs at a frequency of 10 Hz.
- *Config 2:* To generate moderate data traffic, we assume a scenario with 300 neighboring vehicles broadcasting periodic beacons at a frequency of 20 Hz. It is worth noting that Configs 1 and 2 in Table 5 exhibit data densities of 4750 and 1300 packets s^{-1} km^{-1}, respectively, which has a significant impact on communication quality.

We consider the following simulation scenarios for validating the PlatoonSAFE modules:

- *Platoon cruising in a sinusoidal fashion:* The leading vehicle in the platoon oscillates sinusoidally at a frequency of 0.2 Hz and an amplitude of 10 kmh^{-1}. Two simulation scenarios are considered where the platoon employs either the PATH CACC or the PLOEG CACC controller without RTM. Subsequently, we present the simulation results incorporating RTM.
- *Predicted delays in steady-speed platooning:* We consider the steady speed scenario in Figure 5 to validate the integration of the ML models in predicting real-time communication delays.
- *Emergency braking from a steady speed:* Emergency braking is initiated in the platoon at 70 seconds using one of the braking strategies. The simulations are conducted with different inter-vehicle gaps and controllers.

B. IMPACT OF TRANSIENT COMMUNICATION OUTAGES ON PLATOON CRUISING

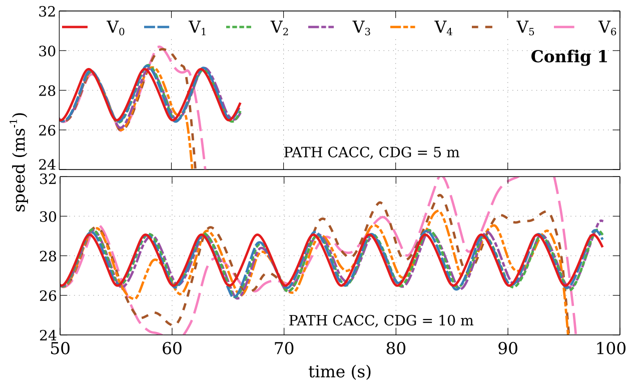
In this subsection, we first evaluate the platoon performance without RTM, i.e., PATH CACC and PLOEG CACC are activated independently under Configs 1 and 2 in Table 5. Next, we evaluate how RTM handles transient communication outages using the default contracts of the RTM module.

1) SINUSOIDAL SCENARIO WITHOUT RTM

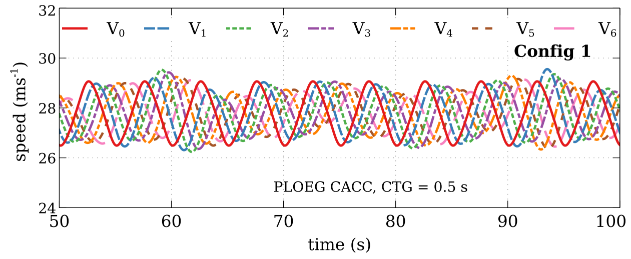
Speed profiles of platoon vehicles using PATH CACC and PLOEG CACC under Config 1 are shown in Figure 6. Figure 6(a) shows speed profiles for inter-vehicle gaps of 5 and 10 m with PATH CACC. The results indicate that rear vehicles experience collisions with a 5 m gap, causing the platoon to stop cruising. Collisions also occur with 10 m gaps around 96 s into the simulation. However, it should be noted that these collisions do not occur in all simulation runs with PATH CACC. Five out of twenty simulation runs with PATH CACC resulted in collisions for

TABLE 6. No. of collisions during cruising under Config 1 for various inter-vehicle gaps and controllers.

Controller	PATH CACC		PLOEG CACC			ACC		
CDGs/ CTGs	5 m	10 m	0.25 s	0.5 s	1.0 s	0.5 s	0.8 s	1.2 s
No. of collisions out of 20 runs	5	5	0	0	0	0	0	0

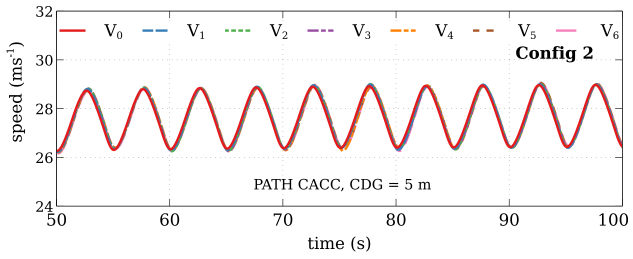


(a) PATH CACC performance evaluated with CDGs of 5 and 10 m under Config 1.

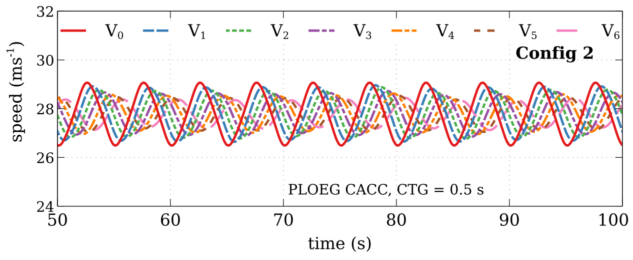


(b) PLOEG CACC evaluated with 0.5 s CTG under Config 1.

FIGURE 6. The speed profiles (ms^{-1}) of platooning vehicles under Config 1 are presented for the PATH CACC and PLOEG CACC controllers without RTM; a 0.5 s CTG corresponds to 15.89 m gap at 100 kmh^{-1}.



(a) PATH CACC with 5 m CDG under Config 2.



(b) PLOEG CACC with 0.5 s CTG under Config 2

FIGURE 7. Speed profiles of platooning vehicles (ms^{-1}) under Config 2 with PATH CACC and PLOEG CACC controllers at lower data density and without RTM; a 0.5 s CTG corresponds to 15.89 m gap at 100 kmh^{-1}.

CDGs of 5 and 10 m, as shown in Table 6. The reason for these collisions is that PATH CACC requires information from both the LV and the preceding vehicle, as illustrated in Figure 3. Communication outages are more prevalent for rear vehicles due to the increasing effects of path loss and fading [2], [3]. During temporary communication outages, vehicles using PATH CACC rely only on the information from the preceding vehicle, leading to the same communication topology as PLOEG CACC, as shown in Figure 3. However, the inter-vehicle gaps are not increased. As a result, the rear vehicles cannot respond to the LV's speed changes fast enough, leading to collisions. However, due to its short gaps, the PATH CACC controller offers high fuel and road efficiency.

Figure 6(b) shows that PLOEG CACC with a 0.5 s CTG successfully avoids collisions in the same simulation scenario as Config 1. Table 6 demonstrates that PLOEG CACC with CTGs of 0.25 s, 0.5 s, and 1.0 s avoids collisions. However, under Config 1, PLOEG CACC cannot attenuate disturbances in the downstream direction and exhibits string instability with a 0.5 s CTG. Additionally, the speed profiles depicted in Figure 6(b) show that the vehicles following PLOEG CACC experience notable delays in tracking the speed changes of the LV. Furthermore, due to the longer inter-vehicle gaps required for safety, PLOEG CACC results in lower fuel and road efficiency compared to PATH CACC.

Figure 7 shows speed profiles of platooning vehicles using PATH CACC and PLOEG CACC controllers under Config 2, where the data density is 1300 packets $s^{-1}km^{-1}$ compared to 4750 packets $s^{-1}km^{-1}$ in Config 1. PATH CACC exhibits safe behavior while enabling high fuel efficiency, road efficiency, and LV tracking ability. PLOEG CACC in Figure 7(b) shows better string stability than in Figure 6(b).

In conclusion, both PATH CACC and PLOEG CACC controllers are significantly impacted by transient communication outages, as shown in Figure 6. However, reducing the data density in Config 2, as seen in Figure 7, drastically improves platoon performance.

2) SINUSOIDAL SCENARIO WITH RTM

Figure 8 presents speed (ms^{-1}) and inter-vehicle distance (m) profiles with RTM enabled under Config 1, i.e., the same data and traffic density as in Figure 6. The RTM uses PATH CACC CDG = 5 m, PLOEG CACC CTG = 0.25 s (8.94 m at 100 kmh^{-1}), and ACC CTG = 0.5 s (15.89 m at

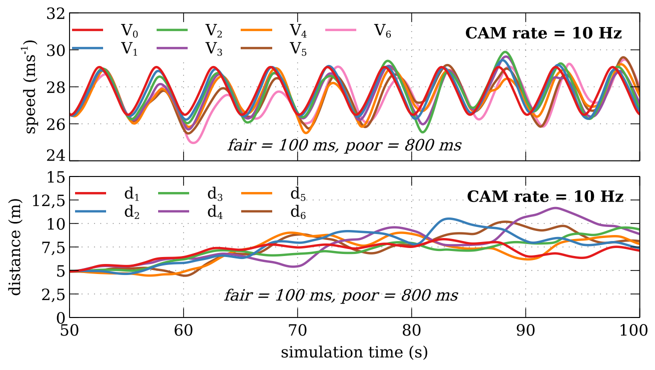


FIGURE 8. Speed (ms^{-1}) and inter-vehicle distance (m) profiles in a sinusoidal scenario using RTM under Config 1; PATH CACC CDG = 5 m, PLOEG CACC CTG = 0.25 s, and ACC CTG = 0.5 s.

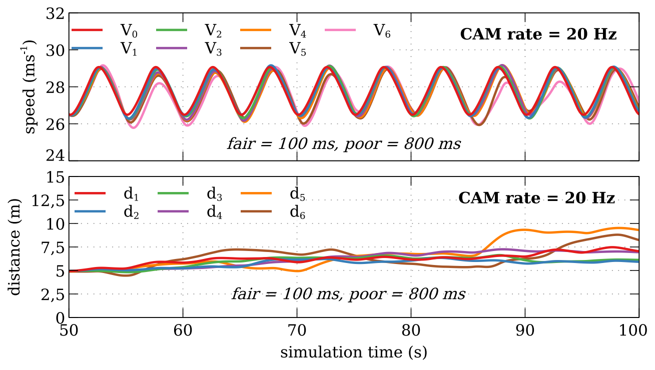


FIGURE 9. Speed (ms^{-1}) and inter-vehicle distance (m) profiles in a sinusoidal scenario using RTM under Config 1; PLATOON CAM rate = 20 Hz, PATH CACC CDG = 5 m, PLOEG CACC CTG = 0.25 s, and ACC CTG = 0.5 s.

100 kmh^{-1}). Recall from Table 2 that a vehicle may increase the gap to the vehicle in front for graceful degradation due to experiencing *fair* communication quality. To this end, simulations in Figure 8 consider a 25% increase in gaps. We assume 100 and 800 ms communication outages as *fair* and *poor* thresholds, respectively. Note that simulations with RTM always begin assuming that communication quality is *good* and the PATH CACC controller with 5 m CDG is active in all the FVs; the LV uses the ACC controller.

The distance profiles shown in Figure 8 indicate that the collisions observed with PATH CACC in Figure 6 can be prevented by incorporating RTM. Furthermore, the speed profiles in Figure 8 reveal that the RTM achieves better string stability and LV tracking performance than both PATH CACC and PLOEG CACC in Figures 6(a) and 6(b), respectively. The distance profiles in Figure 8 can also be used to understand how the RTM instructs the vehicles to switch between different controllers and analyze fuel efficiency and road efficiency. The distance profiles show that the rear vehicles in the platoon temporarily lose string stability due to transient *poor* communication quality with the LV, which leads to adopting the PLOEG CACC controller. As RTM periodically monitors the connectivity, the PATH CACC controller is adopted when a packet is received from the LV. This way, there are frequent switches between the PATH CACC and PLOEG CACC controllers, which leads to frequent changes in inter-vehicle gaps, as seen from the distance profiles in Figure 8. However, the front vehicles are more string stable and able to track the speed changes of the LV, which is crucial during emergency braking. Furthermore, the temporary loss of string stability or increase in inter-vehicle gaps with the rear vehicles is deemed acceptable since RTM degrades the performance to maintain a nominal safety level, which is of utmost importance in the event of temporary failures.

In order to analyze the effects of CAM frequency on platoon performance, we simulate the same scenario as in Figure 8, but with a 20 Hz CAM rate instead of 10 Hz. The speed and inter-vehicle distance profiles are depicted in Figure 9. It is evident that the string stability and LV tracking ability drastically improve with the increase of the CAM rate. Moreover, the distance profiles with a 20 Hz CAM rate demonstrate shorter inter-vehicle gaps than with a lower update rate, enabling higher fuel efficiency and road efficiency. The notable enhancement in platoon performance with higher CAM frequency is due to the faster arrival of repetitions in the event of lost packets; when the CAM rate has doubled, the possibility of receiving CAMs generally increases. It is clear from Figures 6 and 7 that transient communication outages due to packet losses significantly impact platoon performance. With the increase in CAM rates, such packet losses can be alleviated. However, increasing the CAM rate is not always possible, especially in a dense data traffic scenario, due to the restrictions imposed by the Decentralized Congestion Control (DCC) mechanism [79]. Also note that when the data traffic on the communication channel is dense, an increased CAM rate can cause packet collisions which in turn leads to a reduced CAM reception rate as the overall result [80], but in general, increasing the CAM rate increases the redundancy and resilience to packet drops.

The simulation results demonstrate the impacts of wireless communications on platoon safety and performance, as demonstrated by both dense and moderate data traffic scenarios. The RTM module provides a fine-grained model of transient communication outages that allows a platoon to remain fail-operational during cruising by degrading its performance temporarily as a function of communication quality.

C. PREDICTION OF COMMUNICATION DELAYS DURING STEADY SPEED

In this use case, we consider the steady speed of the LV as shown in Figure 5 and carry out real-time prediction of communication delays using the AOSVR and LSTM RNN models. The RMSEs are presented in Table 7. The simulations are performed with CAM rates of 10 Hz and 20 Hz under Config 1. Moreover, in order to estimate the AOSVR parameters ϵ, C_{SVR}, and *SizeLimit*, we conducted simulations for various combinations of the configuration parameters,

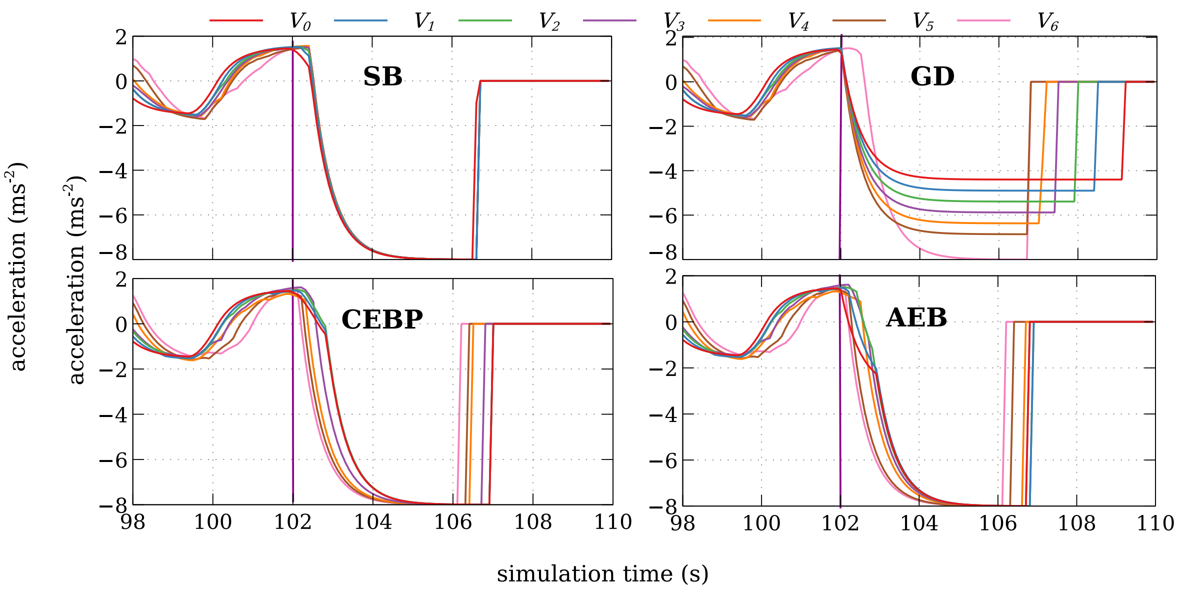


FIGURE 10. Acceleration profiles (ms^{-2}) of the platooning vehicles during an emergency braking at the simulation time 102 s with SB, GD, CEBP, and AEB strategies.

TABLE 7. Root mean square error (RMSE) of predicted delays with ML models.

CAM rate	10 Hz		20 Hz	
ML model	AOSVR	LSTM RNN	AOSVR	LSTM RNN
RMSE	0.327	0.291	0.165	0.153

e.g., no. of lanes and neighboring vehicles, different CAM rates, etc., and evaluated the prediction errors of experienced communication delays. Table 4 shows the estimated parameter values. The simulation results in Table 7 show that LSTM RNN demonstrates lower RMSEs than AOSVR. One possible reason for comparatively higher RMSEs with AOSVR is that the estimated ϵ, C_{SVR}, and *SizeLimit* parameter values are not optimal, which is considered as "big problem in real applications" [66] and requires further investigation.

D. IMPACT OF COMMUNICATION OUTAGES ON EMERGENCY BRAKING

In this subsection, we first present the acceleration profiles of platooning vehicles to validate the implementation of the braking strategies. Next, the impact of wireless communication outages on platoon emergency braking is analyzed in terms of inter-vehicle collisions, time to collision, stopping distance of the leading vehicle, and time to stop the platoon as listed in Section V-B.

1) ACCELERATION PROFILES WITH DIFFERENT BRAKING STRATEGIES

While cruising in a sinusoidal fashion, the LV detects an imaginary road hazard 102 s into the simulation time and starts broadcasting DENMs at 10 Hz. The simulations are carried out under Config 2 in Table 5. The acceleration profiles presented in Figure 10 can be used to understand the correctness of the implementation of the braking strategies. During braking, using the SB strategy, the leading vehicle communicates the waiting period τ_{wait} to the following vehicles (FVs) through DENMs. All the vehicles wait for the event detection time (102 s) + τ_{wait} period before initiating braking at a rate of -8 ms^{-2}. The τ_{wait} period can be calculated by averaging the CAM delays during cruising or predicted using machine learning models. The emergency braking module in PlatoonSAFE provides examples of both methods. Figure 10 shows that the SB strategy facilitates a synchronous braking action of the platooning vehicles. The acceleration profiles with the GD strategy demonstrate that different vehicles brake at different rates, e.g., the LV and the last vehicle brake at rates –4.4 and –8 ms^{-2}. In Figure 10, the acceleration profiles demonstrate that with the CEBP strategy, the last vehicle initiates braking, and the second to last vehicle brakes only after receiving an acknowledgment from the last vehicle. This way, the last vehicle brakes first, and the LV brakes last sequentially at a rate of –8 ms^{-2}. The AEB strategy is similar to CEBP; however, all vehicles except the last vehicle perform soft deceleration at –3 ms^{-2} in Figure 10, while the vehicles are waiting for an ACK from their respective successors. Upon receiving an ACK from the immediate successor, full deceleration is performed at –8 ms^{-2}. The deceleration rates with all the braking strategies can be adjusted in the simulator.

2) EVALUATION OF EMERGENCY BRAKING

In this subsection, we evaluate the NB and SB strategies implemented in the emergency braking module of PlatoonSAFE. To this end, we disable the RTM module during cruising; instead, we consider the PATH CACC and PLOEG CACC controllers with gaps 5 m and 0.4 s (13.1 m at 100 kmh^{-1}), cruising at a steady speed. The PATH CACC represents the case in which information is expected both

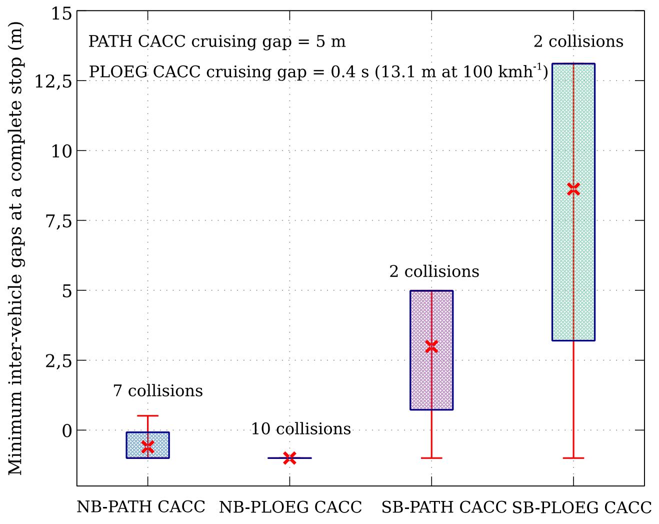


FIGURE 11. Minimum inter-vehicle gaps at a complete standstill (m) after emergency braking with Normal Braking (NB) and Synchronized Braking (SB) strategies under Config 1; cruising speed = 100 kmh^{-1}, deceleration rate = –8 ms^{-2}.

from the LV and the preceding vehicle. In PLOEG CACC, the preceding vehicle is the only source of information. We carried out ten simulation runs for each scenario using Config 1 in Table 5. The platoon cruises at a steady speed of 100 kmh^{-1} following one of the control laws and performs an emergency braking maneuver 70 s into the simulation time at a rate of –8 ms^{-2}.

a) Minimum Inter-vehicle Gaps at a Complete Stop: The minimum gaps between the platooning vehicles after emergency braking, as shown in Figure 11, are presented using box plots. The average of the minimum gaps is denoted by **x**, while the length of the Inter Quartile Range (IQR) indicates the spread of the data points. Using NB, 7 and 10 collision cases out of ten simulation runs can be observed while braking from the cruising states PATH CACC and PLOEG CACC, respectively. Moreover, the IQR below zero with the NB strategy depicts that the minimum inter-vehicle gaps are below zero, i.e., collisions, most of the time. The cause of collisions with PATH CACC is due to long communication delays experienced by the platoon's rear vehicles. If the front vehicle has already received a DENM and started braking, but the ego vehicle experiences communication outages, collisions occur. In addition, the detection, processing, and actuation delays in distance sensors prevent an ego vehicle from responding quickly enough to avoid collisions, particularly when inter-vehicle gaps are short, and the predecessor decelerates at a rate of -8 ms^{-2}. The longer gap with PLOEG CACC is also not adequate for avoiding collisions. The reason is that with longer gaps, the rear vehicles are further away from the LV, causing even more communication outages. Recall from Figure 6(a) and Table 6 that there are collisions during cruising even with longer gaps (10 m) when using PATH CACC. For the same scenario, when SB is used, the inter-vehicle collisions happen in only two out of ten simulation runs; see Figure 11. These collisions with

TABLE 8. Average time to collision (s) from the moment of hazard detection for the same simulations in Figure 11.

Braking strategies	NB		SB	
Controller	PATH CACC	PLOEG CACC	PATH CACC	PLOEG CACC
cruising gaps	5 m	0.4 s	5 m	0.4 s
Time to Collision (s)	2.85	3.81	2.96	4.96

TABLE 9. Average stopping distance of the LV (m) at a complete standstill for the same simulations in Figure 11.

Braking strategies	NB		SB	
Controller	PATH CACC	PLOEG CACC	PATH CACC	PLOEG CACC
cruising gaps	5 m	0.4 s	5 m	0.4 s
Stopping distance (m)	60.82	60.82	73.04	91.93

the SB are mainly caused by the last vehicle in the platoon, which usually experiences the maximum delay. Collisions occur when the last vehicle cannot receive a DENM within the τ_{wait} period, and the preceding vehicle starts braking at a strong deceleration rate. Moreover, the average minimum gaps at a complete standstill with the SB strategy using PATH CACC and PLOEG CACC are around 3 m and 8 m, respectively; these results reveal that the vehicles can perform synchronized braking in most cases due to waiting before braking to mitigate the effects of communication delays.

b) Time to Collision: Table 8 presents the average time to collision for the simulations in Figure 11. Note that this evaluation metric only applies to collision cases with a braking strategy. Furthermore, the results of the first collision in a platoon are presented in Table 8. The results show that a vehicle using PATH CACC undergoes collisions faster than with PLOEG CACC. The reason is that the inter-vehicle distance with PLOEG CACC is 13.1 m compared to the 5 m gap with PATH CACC. As a result, vehicles using the short gaps facilitated by PATH CACC have a shorter time to react in case of emergency braking.

c) Stopping Distance of the LV: Table 9 displays the LV's stopping distance (m) from the moment of encountering the hazard (70 s) until the platoon comes to a complete stop. The stopping distance is shorter with the NB strategy than with the SB strategy. The reason for the longer stopping distance with the SB strategy is that it involves waiting before braking to mitigate the impact of communication delays. However, it is important to consider the LV's stopping distance in addition to collision avoidance when evaluating a braking strategy. Using the NB strategy, the LV would need to brake at a slower deceleration rate to prevent collisions, resulting in a longer stopping distance [48].

d) Time to Stop: Table 10 shows the average time required to bring the entire platoon to a stop from the moment of encountering the simulated hazard. This metric is necessary for evaluating which braking strategy can efficiently transition a platoon to a fail-safe state. As shown in Table 10, the platoon can transition to a fail-safe state more quickly using

TABLE 10. Average time to stop (s) the whole platoon for the same simulations in Figure 11.

Braking strategies	NB		SB	
Controller	PATH CACC	PLOEG CACC	PATH CACC	PLOEG CACC
cruising gaps	5 m	0.4 s	5 m	0.4 s
Time to stop (s)	4.81	7.11	4.48	5.15

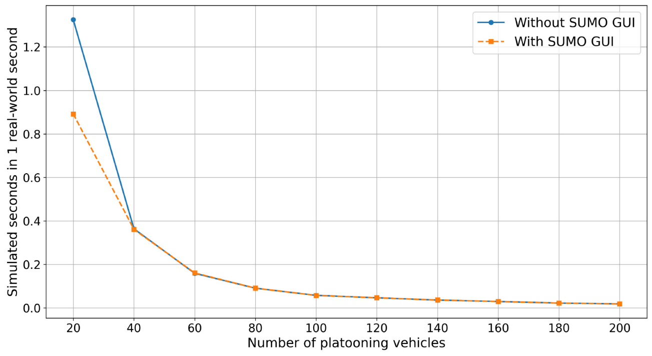


FIGURE 12. Simulation speed of PlatoonSAFE.

the PATH CACC controller than with the PLOEG CACC controller. This is due to the short inter-vehicle gaps used with the PATH CACC controller during cruising.

E. SIMULATION SPEED

The simulation speed of PlatoonSAFE was measured on a laptop with 16 GB RAM, a single CPU (i5-10210U at 1.60GHz), and the Ubuntu 18.04 LTS operating system. The simulation speed is determined by calculating the number of simulated seconds that can be executed within one second of real-world time [15]. The simulation speed of PlatoonSAFE with and without using the SUMO GUI is presented in Figure 12. The X-axis in Figure 12 represents the number of platooning vehicles simulated. For example, 200 platooning vehicles mean that 20 platoons are simulated, each consisting of ten vehicles, and placing them in two separate lanes, with ten platoons in each lane. Figure 12 shows that the simulation of 20 vehicles using GUI can be simulated in real-time using PlatoonSAFE. However, with the increase in the number of platooning vehicles, the number of simulation seconds that can be simulated in one real-world second drastically reduces. The rationale behind the slower simulation speed with PlatoonSAFE is that it is a complex simulator consisting of SUMO, Veins, PLEXE, and PlatoonSAFE features. The Veins simulator, developed using OMNeT++, features a full-fledged network stack with realistic channel models, and its bidirectional coupling with SUMO contributes to the slower simulation speed. However, Zarrad and Alsmadi in [27] show that OMNeT++ is faster in terms of simulation runtime and has lower CPU usage compared to other popular network simulators such as NS-2 and NS-3. Therefore, OMNeT++ is still a reasonable choice for network simulations. Lai et al. [15] show that their generic simulation platform can simulate 216 CAVs in real-time compared to 10 CAVs with the PreScan simulator (a MATLAB/Simulink-based simulator for evaluating Advanced Driver Assistance Systems). However, the simulation speed computed in [15] does not consider sensor and communication models.

VII. POTENTIAL EXTENSIONS OF PLATOONSAFE

PlatoonSAFE enables the evaluation of fail-operational and fail-safe platooning through realistic modeling of transient communication outages. Additionally, the tool integrates two machine learning models that can predict network, communication, and traffic parameters. Researchers from diverse domains can use the simulation tool to consider a realistic communication model that has a significant impact on platoon safety.

The RTM module of PlatoonSAFE includes a switching mechanism that switches between ACC [49], PLOEG CACC [50], and PATH CACC [51] controllers based on the experienced communication quality. While our focus is on validating the modeling of transient communication outages in the longitudinal control of vehicles, it should be noted that in practical scenarios, platoons, CACC-enabled vehicles, or neighboring vehicles must also execute lateral movements such as lane changes, cut-in/cut-out maneuvers, and entry/exit ramp maneuvers. It is possible to use the communication model provided by the RTM module to evaluate lateral control in a realistic manner without compromising the generality of the approach. Readers interested in the possible future extensions of PlatoonSAFE and the evaluation of platoon lateral control under realistic communication scenarios are directed to the relevant literature. For instance, Zhang et al. introduce the Human-Lead-Platoon CACC (HLP-CACC) controller, which enables both longitudinal and lateral control of a platoon with a human-driven automated vehicle as the leader [81]. The study demonstrates that the HLP-CACC controller can mitigate the disturbances caused by the human-driven vehicle and ensure string stability. However, the authors also note that stable communication is a prerequisite for achieving string stability using HLP-CACC. For the purpose of evaluating HLP-CACC in the presence of surrounding traffic, the authors use the generic simulation platform developed by Lai et al. [15], which defines rules for platoon merging and lane-changing maneuvers of CAVs. Wang et al. [82] propose the Making Space CACC Lane Change (MS-CACCLC) controller, which enables a platoon to make space between vehicles in the target lane to accommodate the lane-changing maneuver. In addition, they propose the Successive Platoon Lane Change (SuPLC) controller in [83] to facilitate the overtaking maneuver of a platoon through a small window. Both the MS-CACCLC and SuPLC controllers confirm string stability. Additionally, the SuPLC controller confirms lateral stability, making it useful for both longitudinal and lateral control of CAVs in dense traffic scenarios. Hu et al. propose a set of rules governing CAV behavior in signalized intersection control and scenarios involving high-priority vehicles [16], which are commonly encountered in urban environments. The literature reviewed above presents various approaches for lateral movements of CAVs in realistic road traffic scenarios. In such scenarios, high data traffic generated by neighboring vehicles can cause time-varying communication delays, as shown in Section VI. Therefore, the realistic modeling of transient communication outages in RTM can be valuable in evaluating both longitudinal and lateral control of CAVs. To this end, for instance, the PERMIT simulator [39], developed by Mena-Oreja and Gozalvez extending PLEXE to facilitate merge, split, and leave maneuvers, can be integrated into PlatoonSAFE and further modified to evaluate lateral and longitudinal control of CAVs.

Moreover, PlatoonSAFE can be extended to evaluate fuel efficiency under various traffic scenarios using the RTM module or different CACC controllers. We refer readers to [77] in which fuel consumption is evaluated using the PLEXE simulator. Furthermore, it would be interesting to integrate PlatoonSAFE with Veins-LTE [84] to simulate different teleoperated platoon driving scenarios in which local fault tolerance during runtime is facilitated by PlatoonSAFE, and high-level instructions are received from a remote station.

VIII. CONCLUSION

This paper presents the open-source simulation tool PlatoonSAFE to enable the evaluation of platoon safety under realistic communication, vehicle dynamics, control, and road traffic scenarios. PlatoonSAFE consists of the Runtime Manager (RTM), emergency braking, and Machine Learning (ML) modules. The RTM and the emergency braking modules in PlatoonSAFE can be used together to design how to handle transient communication outages during platoon cruising and emergency braking. To this end, The RTM module models the transient nature of wireless connectivity into finer granularities and enables the evaluation of fail-operational platooning during cruising. Rigorous simulation studies demonstrate that transient communication outages significantly impact platoon safety and performance. The RTM module can enable the attribution of platoon performance levels as a function of communication quality. Furthermore, the simulation results with RTM reveal the necessity of considering the realistic nature of wireless connectivity while evaluating platooning from, e.g., control theory, automotive, or traffic engineering perspectives. In addition, the RTM module allows us to define the communication requirements of different platooning vehicles during emergency braking. Our simulation results demonstrate that platoon safety during an emergency braking depends on the information available during braking, inter-vehicle gaps, speed, deceleration rates, and more. In addition, while evaluating emergency braking strategies, it is not sufficient to consider collision avoidance as the only metric to evaluate braking strategies; the metrics such as the stopping distance of the leading vehicle and time to transition a platoon to a fail-safe state should also be considered. The emergency braking module of PlatoonSAFE provides the implementation of different types of event-driven messages, such as Decentralized Environmental Notification Messages (DENMs), Acknowledgement (ACK), relay messages, and more, that can be used to design how a platoon should behave in case of different types of hazards. In addition, the ML module enables online training of ML models and the real-time prediction of communication parameters. The results demonstrate that communication link quality can be forecasted with decent accuracy when ML models are trained online, and predictions are made in real time. The predictions can be used to design platoon safety applications and provide application-specific Quality of Services (QoS). PlatoonSAFE is made available to the community with documentation to foster the experimentation of different kinds of platoon-related failures and hazards and their mitigation strategies.

REFERENCES

[1] S. E. Shladover, C. Nowakowski, X.-Y. Lu, and R. Ferlis, “Cooperative adaptive cruise control: Definitions and operating concepts,” *Transp. Res. Rec. J. Transp. Res. Board*, vol. 1, no. 2489, pp. 145–152, 2015.

[2] S. Hasan, S. Girs, and E. Uhlemann, “Characterization of transient communication outages into states to enable autonomous fault tolerance in vehicle platooning,” *IEEE Open J. Intell. Transp. Syst.*, vol. 4, pp. 101–129, 2023.

[3] G. Jornod, A. E. Assaad, and T. Kürner, “Packet inter-reception time conditional density estimation based on surrounding traffic distribution,” *IEEE Open J. Intell. Transp. Syst.*, vol. 1, pp. 51–62, 2020.

[4] T. Stolte et al., “Taxonomy to unify fault tolerance regimes for automotive systems: Defining fail-operational, fail-degraded, and fail-safe,” *IEEE Trans. Intell. Veh.*, vol. 7, no. 2, pp. 251–262, Jun. 2022.

[5] K. Bilstrup, E. Uhlemann, E. G. Ström, and U. Bilstrup, “On the ability of the 802.11 p MAC method and STDMA to support real-time vehicle-to-vehicle communication,” *EURASIP J. Wireless Commun. Netw.*, vol. 2009, no. 1, 2009, Art. no. 902414.

[6] L. Cui, J. Hu, B. B. Park, and P. Bujanovic, “Development of a simulation platform for safety impact analysis considering vehicle dynamics, sensor errors, and communication latencies: Assessing cooperative adaptive cruise control under cyber attack,” *Transp. Res. C Emerg. Technol.*, vol. 97, pp. 1–22, Dec. 2018.

[7] K. Li, Y. Bian, S. E. Li, B. Xu, and J. Wang, “Distributed model predictive control of multi-vehicle systems with switching communication topologies,” *Transport. Res. C Emerg. Technol.*, vol. 118, Sep. 2020, Art. no. 102717.

[8] I. Sljivo, B. Gallina, and B. Kaiser, “Assuring degradation cascades of car platoons via contracts,” in *Proc. SAFECOMP*, Magdeburg, Germany, Sep. 2017, pp. 317–329. [Online]. Available: http://www.es.mdh.se/publications/4787-

[9] J. S. Weber, M. Neves, and T. Ferreto, “VANET simulators: An updated review,” *J. Braz. Comput. Soc.*, vol. 27, no. 1, pp. 1–31, 2021.

[10] S. Hasan, “Fail-operational and fail-safe vehicle platooning in the presence of transient communication errors,” Mälardalen Univ., Västerås, Sweden, Rep. 1878487, Mar. 2022.

[11] J. Almeida, J. Rufino, M. Alam, and J. Ferreira, “A survey on fault tolerance techniques for wireless vehicular networks,” *Electronics*, vol. 8, no. 11, p. 1358, 2019. [Online]. Available: https://www.mdpi.com/2079-9292/8/11/1358

[12] L. Torres-Figueroa, H. F. Schepker, and J. Jiru, “QoS evaluation and prediction for C-V2X communication in commercially-deployed LTE and mobile edge networks,” in *Proc. IEEE 91st Veh. Technol. Conf. (VTC-Spring)*, 2020, pp. 1–7.

[13] S. Barmpounakis et al., “AI-driven, QoS prediction for V2X communications in beyond 5G systems,” *Comput. Netw.*, vol. 217, Nov. 2022, Art. no. 109341. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128622003759

[14] W. Zhang, M. Feng, M. Krunz, and H. Volos, "Latency prediction for delay-sensitive V2X applications in mobile cloud/edge computing systems," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2020, pp. 1–6.

[15] J. Lai, J. Hu, L. Cui, Z. Chen, and X. Yang, "A generic simulation platform for cooperative adaptive cruise control under partially connected and automated environment," *Transp. Res. C Emerg. Technol.*, vol. 121, Dec. 2020, Art. no. 102874.

[16] J. Hu, S. Sun, J. Lai, S. Wang, Z. Chen, and T. Liu, "CACC simulation platform designed for urban scenes," *IEEE Trans. Intell. Veh.*, early access, Jan. 6, 2023, doi: 10.1109/TIV.2023.3234890.

[17] R. Rajamani and S. Shladover, "An experimental comparative study of autonomous and co-operative vehicle-follower control systems," *Transp. Res. C Emerg. Technol.*, vol. 9, no. 1, pp. 15–31, 2001.

[18] B. van Arem, C. J. G. van Driel, and R. Visser, "The impact of cooperative adaptive cruise control on traffic-flow characteristics," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 4, pp. 429–436, Dec. 2006.

[19] C. Sommer, R. German, and F. Dressler, "Bidirectionally coupled network and road traffic simulation for improved IVC analysis," *IEEE Trans Mobile Comput.*, vol. 10, no. 1, pp. 3–15, Jan. 2011.

[20] M. Segata, S. Joerer, B. Bloessl, C. Sommer, F. Dressler, and R. Lo Cigno, "PLEXE: A Platooning extension for veins," in *Proc. IEEE VNC*, Paderborn, Germany, Dec. 2014, pp. 53–60.

[21] *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specification of Decentralized Environmental Notification Basic Service, V1.2.1*, ETSI Standard EN 302 637-3, Nov. 2014.

[22] D. Jia, K. Lu, J. Wang, X. Zhang, and X. Shen, "A survey on platoon-based vehicular cyber-physical systems," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 263–284, 1st Quart., 2016.

[23] P. A. Lopez et al., "Microscopic traffic simulation using SUMO," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, 2018, pp. 2575–2582.

[24] M. Fiore, J. Harri, F. Filali, and C. Bonnet, "Vehicular mobility simulation for VANETs," in *Proc. 40th Annu. Simulat. Symp. (ANSS)*, 2007, pp. 301–309.

[25] M. Fellendorf, "VISSIM: A microscopic simulation tool to evaluate actuated signal control including bus priority," in *Proc. 64th Inst. Transp. Eng. Annu. Meeting*, vol. 32, 1994, pp. 1–9.

[26] A. Varga, "The OMNET++ discrete event simulation system," in *Proc. ESM*, Jun. 2001, pp. 1–8.

[27] A. Zarrad and I. Alsmadi, "Evaluating network test scenarios for network simulators systems," *Int. J. Distrib. Sensor Netw.*, vol. 13, no. 10, p. 19, 2017.

[28] A. Wegener, M. Piórkowski, M. Raya, H. Hellbrück, S. Fischer, and J.-P. Hubaux, "TraCI: An interface for coupling road traffic and network simulators," in *Proc. CNS*, 2008, pp. 155–163.

[29] M. Piórkowski, M. Raya, A. L. Lugo, P. Papadimitratos, M. Grossglauser, and J.-P. Hubaux, "TraNS: Realistic joint traffic and network simulator for VANETs," *SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 12, no. 1, pp. 31–33, Jan. 2008. [Online]. Available: https://doi.org/10.1145/1374512.1374522

[30] M. Rondinone et al., "iTETRIS: A modular simulation platform for the large scale evaluation of cooperative ITS applications," *Simulat. Model. Pract. Theory*, vol. 34, pp. 99–125, May 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1569190X1300018X

[31] *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service, V1.3.1*, ETSI Standard EN 302 637-2, Nov. 2014.

[32] B. van Arem, A. De Vos, and M. J. Vanderschuren, "The microscopic traffic simulation model MIXIC 1.3," Dept. Traffic Transp., TNO Intro, Delft, The Netherlands, TNO Rep. INRO-VVG 1997-02b, 1997.

[33] C. Lei, E. M. van Eenennaam, W. K. Wolterink, G. Karagiannis, G. Heijenk, and J. Ploeg, "Impact of packet loss on CACC string stability performance," in *Proc. ITST*, 2011, pp. 381–386.

[34] F. Gechter, J.-M. Contet, S. Galland, O. Lamotte, and A. Koukam, "Virtual intelligent vehicle urban simulator: Application to vehicle platoon evaluation," *Simulat. Model. Pract. Theory*, vol. 24, pp. 103–114, May 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1569190X12000172

[35] M. Guériau, B. Dafflon, and F. Gechter, "VIPS: A simulator for platoon system evaluation," *Simulat. Model. Pract. Theory*, vol. 77, pp. 157–176, Sep. 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1569190X17300904

[36] L. Zhao and J. Sun, "Simulation framework for vehicle platooning and car-following behaviors under connected-vehicle environment," *Procedia Soc. Behav. Sci.*, vol. 96, pp. 914–924, Nov. 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877042813022313

[37] A. Choudhury et al., "An integrated V2X simulator with applications in vehicle platooning," in *Proc. ITSC*, Rio, Brazil, 2016, pp. 1017–1022.

[38] R. Rajamani, *Vehicle Dynamics and Control*. New York, NY, USA: Springer, 2011.

[39] J. Mena-Oreja and J. Gozalvez, "Permit—A SUMO simulator for platooning maneuvers in mixed traffic scenarios," in *Proc. ITSC*, 2018, pp. 3445–3450.

[40] M. Amoozadeh, B. Ching, C.-N. Chuah, D. Ghosal, and H. M. Zhang, "VENTOS: Vehicular network open simulator with hardware-in-the-loop support," *Procedia Comput. Sci.*, vol. 151, pp. 61–68, 2019.

[41] M. Malik, M. Maleki, P. Folkesson, B. Sangchoolie, and J. Karlsson, "ComFASE: A tool for evaluating the effects of V2V communication faults and attacks on automated vehicles," in *Proc. DSN*, Baltimore, MD, USA, 2022, pp. 185–192.

[42] F. D. Vita, D. Bruneo, A. Puliafito, G. Nardini, A. Virdis, and G. Stea, "A deep reinforcement learning approach for data migration in multi-access edge computing," in *Proc. ITU Kaleidoscope Mach. Learn. 5G Future (ITU K)*, 2018, pp. 1–8.

[43] A. Virdis, G. Stea, and G. Nardini, "Simulating LTE/LTE-advanced networks with SimuLTE," in *Simulation and Modeling Methodologies, Technologies and Applications*. Cham, Switzerland: Springer, 2015, pp. 83–105.

[44] M. Segata et al., "Multi-technology cooperative driving: An analysis based on PLEXE," *IEEE Trans. Mobile Comput.*, early access, Feb. 25, 2022, doi: 10.1109/TMC.2022.3154643.

[45] A. Memedi, C. Tebruegge, J. Jahneke, and F. Dressler, "Impact of vehicle type and headlight characteristics on vehicular VLC performance," in *Proc. IEEE Veh. Netw. Conf. (VNC)*, 2018, pp. 1–8.

[46] H. Liu, X. D. Kan, S. E. Shladover, X.-Y. Lu, and R. E. Ferlis, "Modeling impacts of cooperative adaptive cruise control on mixed traffic flow in multi-lane freeway facilities," *Transp. Res. C Emerg. Technol.*, vol. 95, pp. 261–279, Oct. 2018.

[47] L. Xiao, M. Wang, and B. van Arem, "Realistic car-following models for microscopic simulation of adaptive and cooperative adaptive cruise control vehicles," *Transp. Res. Rec.*, vol. 2623, no. 1, pp. 1–9, 2017.

[48] S. Hasan, A. Balador, S. Girs, and E. Uhlemann, "Towards emergency braking as a fail-safe state in platooning: A simulative approach," in *Proc. IEEE VTC-Fall*, Honolulu, HI, USA, 2019, pp. 1–5.

[49] P. A. Ioannou and C. C. Chien, "Autonomous intelligent cruise control," *IEEE Trans. Veh. Technol.*, vol. 42, no. 4, pp. 657–672, Nov. 1993.

[50] J. Ploeg, B. T. M. Scheepers, E. van Nunen, N. van de Wouw, and H. Nijmeijer, "Design and experimental evaluation of cooperative adaptive cruise control," in *Proc. ITSC*, Washington, DC, USA, Oct. 2011, pp. 260–265.

[51] R. Rajamani, H.-S. Tan, B. K. Law, and W.-B. Zhang, "Demonstration of integrated longitudinal and lateral control for the operation of automated vehicles in platoons," *IEEE Trans. Control Syst. Technol.*, vol. 8, no. 4, pp. 695–708, Jul. 2000.

[52] M. Segata, "Safe and efficient communication protocols for platooning control," Ph.D. dissertation, Dept. Comput. Sci., Univ. Innsbruck, Tyrol, Austria, Feb. 2016.

[53] S. Hasan, M. A. A. Ahad, I. Sljivo, A. Balador, S. Girs, and E. Lisova, "A fault-tolerant controller manager for platooning simulation," in *Proc. IEEE ICCVE*, Graz, Austria, 2019, pp. 1–6.

[54] B. Kaiser, B. Monajemi, D. Kusche, and H. Schulte, "Systematic design and validation of degradation cascades for safety-relevant systems," in *Proc. ICEST*, Jun. 2017, p. 527.

[55] B. Kaiser, R. Weber, M. Oertel, E. Böde, B. M. Nejad, and J. Zander, "Contract-based design of embedded systems integrating nominal behavior and safety," in *Proc. CSIMQ*, Oct. 2015, pp. 66–91.

[56] I. Sljivo, B. Gallina, J. Carlson, and H. Hansson, "Strong and weak contract formalism for third-party component reuse," in *Proc. ISSREW*, 2013, pp. 359–364.

[57] W. Yu, X. Hua, and W. Wang, "Investigating the longitudinal impact of cooperative adaptive cruise control vehicle degradation under communication interruption," *IEEE Intell. Transp. Syst. Mag.*, vol. 14, no. 4, pp. 183–201, Jul./Aug. 2022.

[58] J. Ploeg, E. Semsar-Kazerooni, G. Lijster, N. van de Wouw, and H. Nijmeijer, "Graceful degradation of cooperative adaptive cruise control," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 488–497, Feb. 2015.

[59] S. Girs, I. Sljivo, and O. Jaradat, "Contract-based assurance for wireless cooperative functions of vehicular systems," in *Proc. 43rd Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Beijing, China, 2017. [Online]. Available: http://www.es.mdh.se/publications/4817

[60] A. Fermi, M. Mongelli, M. Muselli, and E. Ferrari, "Identification of safety regions in vehicle platooning via machine learning," in *Proc. 14th IEEE Int. Workshop Factory Commun. Syst. (WFCS)*, 2018, pp. 1–4.

[61] I. Cara and J.-P. Paardekooper, "The potential of applying machine learning for predicting cut-in behavior of surrounding traffic for truck-platooning safety," in *Proc. 25th Int. Technol. Conf. Enhanced Safety Veh. (ESV)*, Jun. 2017, pp. 1–8.

[62] M. I. Khan, F.-X. Aubet, M.-O. Pahl, and J. Härri, "Deep learning-aided application scheduler for vehicular safety communication," 2019, *arXiv:1901.08872*.

[63] S. S. Sepasgozar and S. Pierre, "Network traffic prediction model considering road traffic parameters using artificial intelligence methods in VANET," *IEEE Access*, vol. 10, pp. 8227–8242, 2022.

[64] F. Jaffar, T. Farid, M. Sajid, Y. Ayaz, and M. J. Khan, "Prediction of drag force on vehicles in a platoon configuration using machine learning," *IEEE Access*, vol. 8, pp. 201823–201834, 2020.

[65] Z. M. Fadlullah et al., "State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2432–2455, 4th Quart., 2017.

[66] F. Parrella, "Online support vector machines for regression," M.S. thesis, Inf. Sci., Dept. Inf. Sci., Univ. Genoa, Genoa, Italy, 2007.

[67] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Comput.*, vol. 31, no. 7, pp. 1235–1270, 2019.

[68] F. Tang, B. Mao, N. Kato, and G. Gui, "Comprehensive survey on machine learning in vehicular network: Technology, applications and challenges," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 2027–2057, 3rd Quart., 2021.

[69] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Rep.*, Dec. 2014, pp. 1–9.

[70] X. Liu, A. Goldsmith, S. S. Mahal, and J. K. Hedrick, "Effects of communication delay on string stability in vehicle platoons," in *Proc. ITSC*, 2001, pp. 625–630.

[71] R. Zheng, K. Nakano, S. Yamabe, M. Aki, H. Nakamura, and Y. Suda, "Study on emergency-avoidance braking for the automatic platooning of trucks," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 4, pp. 1748–1757, Aug. 2014.

[72] C. Bergenhem, K. Meinke, and F. Ström, "Quantitative safety analysis of a coordinated emergency brake protocol for vehicle platoons," in *Proc. ISoLA*, 2018, pp. 386–404.

[73] L. Cheng, B. E. Henty, D. D. Stancil, F. Bai, and P. Mudalige, "Mobile vehicle-to-vehicle narrow-band channel measurement and characterization of the 5.9 GHz dedicated short range communication (DSRC) frequency band," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 8, pp. 1501–1516, Oct. 2007.

[74] C. Sommer, S. Joerer, and F. Dressler, "On the applicability of two-ray path loss models for vehicular network simulation," in *Proc. IEEE Veh. Netw. Conf. (VNC)*, 2012, pp. 64–69.

[75] C. Sommer, D. Eckhoff, R. German, and F. Dressler, "A computationally inexpensive empirical model of IEEE 802.11p radio shadowing in urban environments," in *Proc. 8th Int. Conf. Wireless On-Demand Netw. Syst. Services*, 2011, pp. 84–90.

[76] M. Segata et al., "Toward communication strategies for platooning: Simulative and experimental evaluation," *IEEE Trans. Veh. Technol.*, vol. 64, no. 12, pp. 5411–5423, Dec. 2015.

[77] N. Lyamin, Q. Deng, and A. Vinel, "Study of the platooning fuel efficiency under ETSI ITS-G5 communications," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, 2016, pp. 551–556.

[78] G. Jornod, A. Pfadler, S. Carreira, A. E. Assaad, and T. Kürner, "Fuel efficient high-density Platooning using future conditions prediction," *IEEE Open J. Intell. Transp. Syst.*, vol. 3, pp. 786–798, 2022.

[79] G. Bansal, J. B. Kenney, and C. E. Rohrs, "LIMERIC: A linear adaptive message rate algorithm for DSRC congestion control," *IEEE Trans. Veh. Technol.*, vol. 62, no. 9, pp. 4182–4197, Nov. 2013.

[80] A. Böhm, M. Jonsson, and E. Uhlemann, "Co-existing periodic beaconing and hazard warnings in IEEE 802.11p-based platooning applications," in *Proc. 10th ACM Int. Workshop Veh. Inter Netw. Syst. Appl. (VANET)*, 2013, pp. 99–102.

[81] Y. Zhang et al., "Human-lead-platooning cooperative adaptive cruise control," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 18253–18272, Oct. 2022.

[82] H. Wang, J. Lai, X. Zhang, Y. Zhou, S. Li, and J. Hu, "Make space to change lane: A cooperative adaptive cruise control lane change controller," *Transp. Res. C Emerg. Technol.*, vol. 143, Oct. 2022, Art. no. 103847.

[83] H. Wang, X. Li, X. Zhang, J. Hu, X. Yan, and Y. Feng, "Cut through traffic like a snake: Cooperative adaptive cruise control with successive platoon lane-change capability," *J. Intell. Transp. Syst.*, to be published.

[84] F. Hagenauer, F. Dressler, and C. Sommer, "Poster: A simulator for heterogeneous vehicular networks," in *Proc. IEEE Veh. Netw. Conf. (VNC)*, 2014, pp. 185–186.

SHAHRIAR HASAN (Student Member, IEEE) received the B.Sc. degree in computer science and engineering from the Islamic University of Technology, Dhaka, Bangladesh, in 2013, and the joint master's degree in Internet technologies and architecture from Sorbonne Université, Paris, France, and the University of Trento, Italy, in 2017. He is currently pursuing the Ph.D. degree in computer science and engineering with Mälardalen University, Västerås, Sweden.

From 2013 to 2015, he was a Software Engineer with Samsung R&D Institute Bangladesh Ltd., Dhaka. His research interests include connected vehicles, wireless vehicular communications, fault-tolerance methods in safety-critical systems, software-defined networking, network function virtualization, 5G radio access networks, and network simulation and modeling.

JOSEBA GOROSPE received the B.Sc. degree in computer engineering and telecommunication systems engineering from Mondragon Unibertsitatea, Arrasate, Spain, in 2019, and the M.Sc. degree in computational engineering and intelligent systems from Euskal Herriko Unibertsitatea, Donostia, Spain, in 2020. He is currently pursuing the Ph.D. degree with Mondragon Unibertsitatea, focusing his research area on intelligent transport systems. He completed his bachelor thesis in Ormazabal from 2018 to 2019 and his master thesis in Tecnalia Research & Innovation in 2020. He worked as an Intern with the Signal Theory and Communications Group, Mondragon Unibertsitatea from 2016 to 2018. His research interests are vehicular communications, artificial intelligence applied to ITS, and mobile-edge computing.

SVETLANA GIRS (Member, IEEE) received the B.Sc. and M.Sc. degrees in telecommunications from Saint-Petersburg State Polytechnic University, Russia, in 2009 and 2011, respectively, and the Ph.D. degree in computer science and engineering from Mälardalen University, Västerås, Sweden, in February 2016, where She is a Senior Lecturer with the School of Innovation, Design and Engineering and is currently leading the Networked and Embedded Systems Division. She has been a Visiting Researcher with the University of Canterbury, Christchurch, New Zealand, for three months in 2014. Her research interests include real-time communication, cooperative relay networks and reliable wireless communication for industrial systems, and vehicular communication. She is also a Co-Chair of the Subcommittee on Industrial Communication Systems within the IEEE IES Technical Committee on Factory Automation.

ARRATE ALONSO GÓMEZ was born in Bilbao (Basque Country), Spain, in 1985. She received the M.Sc. degree in Telecommunications engineering from ESIDE, University of Deusto, Bilbao, in 2008, the M.Sc. degree in communication technology, systems and networks (specialized in signal theory) from the Universidad Politécnica de Valencia, Valencia, Spain, in 2009, and the Dr.-Tech. degree (with Hons.), with a Ph.D. thesis on medium access-control layer protocol design (optimization) for vehicular communications, specifically for delay-sensitive and safety-related applications in 2013. Then, she joined the MOBI Group (MOBI Mobility, Logistics and Automotive Technology Research Center), Vrije Universiteit Brussel, Brussels, Belgium, as a Postdoctoral Researcher in connected and electric mobility strategies. She participated in the European Coordination Action Smart EV-VC and coordinated the EU Project GO4SEM. Since September 2015, she has been a Researcher and a Lecturer with EPS-MU, Arrasate, Spain, where she is a member of the Signal Theory and Communications Research Team, Computing and Electronics Department. At EPS-MU, she has been a Main Researcher of the SCOTT Project (ECSEL-2016 the AUTOLIB (Elkartek 2019) Project), where the performance of ITS-G5 is/has been evaluated for Autonomous Train Operation. She is currently a Main Researcher of the InSecTT Project (ECSEL-2019), the AUTOEV@L (Elkartek 2021) projects, where the performance of ITS-G5, C-V2X, and 802.11bd Multi-Radio Access Technologies is evaluated for Intelligent Automation Services for Smart Transportation scenarios.

ELISABETH UHLEMANN (Senior Member, IEEE) received the Ph.D. degree in communications theory from the Chalmers University of Technology, Gothenburg, Sweden, in 2004. She has held visiting positions with the University of South Australia in 2005, the Technical University of Berlin in 2007, and the University of Canterbury, New Zealand, in 2011. She has also worked as a Consultant with Volvo Technology from 2005 to 2009 dealing with connected vehicles, with Ikanos Communications, Fremont, CA, USA, in 2005 with VDSL protocols, and with Free2move, Sweden, from 2009 to 2010 with wireless audio. She is currently a Full Professor of Data Communications with Mälardalen University, Sweden. She has been involved in several European projects studying communication requirements for traffic-safety applications in vehicular networks. She also contributed to the European ITS communications architecture and has served as a Technical Expert in standardization within ETSI TC ITS. She currently serves as a Senior Editor for *IEEE Vehicular Technology Magazine* on Connected and Automated Vehicles and has served as a member of different Ph.D. examination committees over 20 times in five different countries. She has served as the Vice Chair of the Swedish VT/COM/IT Chapter for several years.