Received 1 February 2025; revised 10 March 2025; accepted 24 March 2025. Date of publication 1 April 2025; date of current version 18 April 2025.

Digital Object Identifier 10.1109/OJCOMS.2025.3556513

A TSN-Like Slot-Based Scheduler for Improved Wireless Quality and Platoon Formation in Smart Factories

ANGELO FERAUDO (Graduate Student Member, IEEE), ANDREA GARBUGLI (Member, IEEE), AND PAOLO BELLAVISTA (Senior Member, IEEE)

(Special Issue on Industrial Communication Networks (ICNets) for Industry 5.0)

Department of Computer Science and Engineering, University of Bologna, 40126 Bologna, Italy

CORRESPONDING AUTHOR: A. FERAUDO (e-mail: angelo.feraudo@unibo.it)

This work was supported in part by the European Union's NextGenerationEU Instrument, under the Italian National Recovery and Resilience Plan (NRRP), Mission 4 Component 2 Investment 1.3, Enlarged Partnership on "Telecommunications of the Future" (Program "RESTART")–Industrial Networks under Grant PE00000001, and in part by the National PRIN JOULE Project.

ABSTRACT The platooning of mobile robots, facilitated by Device-to-Device (D2D) communications, has become central in Industry 4.0, enhancing material transport, reducing energy consumption, and improving safety in smart factories. However, as the number of participating robots increases, maintaining effective communication and coordination becomes increasingly challenging. Shared spectrum usage in D2D networks can lead to communication collisions, particularly in highly dynamic scenarios, posing significant challenges in industrial environments. This paper introduces a novel slot-based solution inspired by Time-Sensitive Networking (TSN) to address these challenges, ensuring reliable, low-latency communication while optimizing resource allocation and communication efficiency in platooning systems. To achieve this, we propose a controller named TSNCtl, operating at the application layer of the network stack. TSNCtl leverages a finite state machine (FSM) to manage platoon formation and employs slot-based scheduling for efficient message dissemination. Using the OMNeT++ simulation framework and INET library, we demonstrate the effectiveness of TSNCtl in reducing packet collisions across a variety of scenarios. Our experimental results highlight a significant improvement over the traditional CSMA/CA baseline employed in Wi-Fi-based protocols such as IEEE 802.11p. For instance, TSNCtl achieves near-zero packet collisions with appropriate configurations, even in highly dynamic environments.

INDEX TERMS TSN, Industry 4.0, AGVs, Wi-Fi, 5G.

I. INTRODUCTION

THE MANUFACTURING industry has undergone significant changes due to technological advances and the rise of Industry 4.0, which integrates physical systems with digital technologies to create Smart Manufacturing Systems (SMSs). An SMS can improve product lifecycle management, enhance operational efficiency, and increase responsiveness to disruptions by connecting various devices. In this context, effective trajectory planning and scheduling of Automated Guided Vehicles (AGVs) is essential to optimize production performance, as optimal coordination of AGVs can maintain the continuous flow of materials and products in a dynamic manufacturing environment [1]. For these reasons, along with the emergence of Industry 5.0 and the increasing need for collaboration between humans and collaborative robots (cobots), the platooning of AGVs and cobots—where multiple vehicles or robots move in a coordinated formation to improve efficiency and reduce delays—has become essential in SMSs [2], [3]. Platooning tackles significant challenges in smart factory settings by enhancing the transport of goods and materials, lowering energy use, and boosting operational safety [3], [4], [5]. Device-to-device (D2D) technology can be harnessed in such a context to create fleets of mobile robots, ensuring accurate material delivery

by enabling direct communication between devices without relying on static network infrastructure. Similarly to smart factory contexts, in Intelligent Transportation Systems (ITSs), platooning and D2D communications can be utilized to form cooperative vehicle groups on the road, unified by shared goals like optimizing traffic flow and providing infotainment services [6], [7].

Central to the platoon concept are the communication protocols. An early initiative to connect vehicles was the introduction of the IEEE 802.11p [8] protocol, which was standardized in the EU within the ITS-G5 framework [9] and in the U.S. under the Wireless Access in Vehicular Environment (WAVE) framework [10]. Additionally, the 3rd Generation Partnership Project (3GPP) standardized existing cellular network frameworks to create a cohesive solution for vehicle-to-everything (V2X) and vehicle-to-vehicle (V2V) communications [11], [12], which have gained traction in industrial applications as well [3].

Indeed, in industrial scenarios 5G sidelink emerges as a groundbreaking feature that facilitates direct D2D communication without centralized infrastructure. It offers two modes of operation: mode 1, where a base station manages sidelink spectrum resources, and mode 2, where devices independently allocate resources by monitoring channel information from nearby devices. This adaptability improves the scalability and resilience of industrial and vehicular networks, positioning D2D communication as a foundational element for future connected environments.

Despite the progress in D2D communication protocols, many still share the same spectrum, leading to potential message collisions. Additionally, as the number of devices increases, maintaining effective communication and coordination among nodes becomes more complex [2], [3], [5], [13], [14]. This complexity increases in industrial contexts, which require deterministic and low-latency communication [15] and often feature periodic message transmission, which can lead to *persistent communication collisions* [16]. Tackling these challenges requires the creation of effective collision avoidance strategies designed specifically for D2D communication settings.

We claim that Time-Sensitive Networking (TSN) principles can be extended to wireless scenarios [17] introducing deterministic and low latency communications in D2D environments [18], [19]. TSN offers precise timing synchronization and deterministic communication capabilities. Hence, leveraging TSN-based slotted protocols and exploiting time synchronization mechanisms makes it possible to orchestrate coordinated message transmission schedules, mitigating the risk of message collisions and enhancing overall communication reliability.

In this paper, we introduce a TSN-like controller, named TSNCtl, designed to facilitate data delivery in mobile node platooning based on D2D communications. For enhanced flexibility and dynamic deployment, TSNCtl operates at the application layer and employs a finite state machine (FSM) to manage platoon formation and message dissemination for intra-platoon communication. The platoon creation and joining process is triggered by receiving application/service-related messages or detecting proximity to an existing platoon. Once a mobile node joins a platoon, TSNCtl intercepts each message generated by the service running on the mobile node, determines the appropriate priority queue for its storage, and disseminates the messages based on allocated time slots and their assigned priority. TSNCtl dynamically adapts message dissemination as nodes join and leave the platoon, ensuring consistent and reliable communication despite the dynamic topology.

To validate the effectiveness of our approach, we implemented TSNCtl on the OMNeT++ simulation platform, leveraging the INET 4.5 framework to replicate D2D communication scenarios using a Wi-Fi-based protocol (IEEE 802.11p). A series of experiments were conducted to evaluate performance metrics, focusing primarily on packet collisions resulting from vehicle communications within a platoon. The results demonstrate that TSNCtl outperforms traditional collision avoidance mechanisms in IEEE 802.11 protocols, such as Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), by significantly reducing packet collisions and improving communication reliability. Additionally, we assessed the robustness of TSNCtl in three scenarios characterized by varying levels of dynamic conditions, thereby showcasing its adaptability and reliability in diverse operational environments.

In summary, the key original contributions of this paper are as follows:

- We propose TSNCtl, a TSN-like controller, to improve data delivery and platoon formation.
- We implement TSNCtl on OMNeT++ with INET 4.5, simulating D2D communication with IEEE 802.11p, and evaluate its performance by comparing packet collisions with those generated by traditional IEEE 802.11 collision avoidance mechanisms.
- We demonstrate TSNCtl's robustness under dynamic conditions, showcasing its adaptability and reliability across diverse operational environments.

The rest of the paper is structured as follows. Section II provides essential background information on TSN and the simulation tools utilized for testing our controller. Section III examines current strategies addressing intra-platoon communication issues. Following that, Section IV details our original TSNCtl controller. We assess our proposal in Section V, concentrating on packet collisions within intra-platoon communications and the resilience of our controller amidst dynamic collisions. In conclusion, we discuss the existing limitations of our approach and the principal extensions that are the focus of our ongoing research efforts.

II. BACKGROUND

This section presents the TSN standard protocol suite, which is widely used for deterministic and low-latency data exchange in wired networks. It also explores how wireless communication can benefit from adopting some

protocols present in the standard. Furthermore, it delves into the simulation tools employed to model complex network behaviors, focusing on the OMNeT++ framework. This versatile simulation platform facilitates the assessment of performance metrics under various network conditions.

A. TIME-SENSITIVE NETWORKING

TSN comprises a suite of standards designed to make Ethernet networks deterministic, mainly to accommodate real-time traffic flows.

A crucial component in applications using TSN protocols is establishing a reliable time synchronization mechanism that ensures all communication entities operate based on a unified time reference. This synchronization is facilitated within TSNs by the IEEE 802.1AS protocol, an extension of the IEEE 1588 Precision Time Protocol (PTP) known as generic PTP (gPTP) [20]. Central to the PTP specification is the concept of a grandmaster clock, typically a highly accurate and stable source such as a Global Positioning System (GPS) receiver. This grandmaster clock serves as the central provider of reference time for all devices in the network. In the context of gPTP, devices in the network are designated as Clock Master (CM) and Clock Slave (CS), each participating in the exchange of synchronization messages. The CM disseminates time information to the connected CSs via multicast communication. Then, each CS, known as a gPTP instance, adjusts its synchronized time, considering the message propagation delay along the gPTP communication path from the leader to the instance. Once synchronization is achieved among all devices, a cohesive and time-aware network emerges, forming the so-called gPTP domain.

For Quality of Service (QoS), the TSN working group defines different traffic shaping techniques [17]. In particular, IEEE 802.1Qbv introduces a time shaper known as the Time-Aware Shaper (TAS). The TAS orchestrates network frame scheduling for different types of time-critical flows [21]. The standard outlines time-aware communication windows associated with a specific transmission queue. These windows are further segregated into cyclically repeating time slots, allowing frames to be buffered until the next associated time slot is opened. This segregation ensures that assured traffic maintains low latency and minimal jitter while avoiding interference from other traffic streams. A Gate Control List (GCL) facilitates the definition of windows and slots, identifying the time instants at which queues are open for frame transmission. As a result, TAS can meet the requirements of ultra-low latency and reliability, provided that all time windows are synchronized, which makes its combination with PTP necessary.

In automotive Ethernet communications, TSN outlines a specialized profile known as IEEE 802.1DG, tailored to meet the rigorous requirements of in-vehicle networks (IVNs) regarding high reliability and low latency [22]. The 802.1DG profile was developed to provide a set of mechanisms that ensure the timely delivery of crucial traffic, particularly control and security messages, while supporting best-effort traffic.

Even if TSN primarily targets wired environments, in the last few years it has been extended to wireless scenarios, especially over IEEE 802.11 with the advent of Wi-Fi 6 and Wi-Fi 7 standards [19], [23]. The IEEE 802.1AS time synchronization standard has been defined to operate over the 802.11 Timing Measurement (TM) protocol and updated later to operate over the 802.11 Fine Timing Measurement (FTM) protocol and can used by TSN-capable networks to provide a single time reference for LAN and Wireless LAN segment. For traffic shaping, it is possible to apply the 802.1Qbv schedule at the network stack of 802.11 devices to shape the traffic above the 802.11 Medium Access Control (MAC) [18], [19]. In this mode, the 802.1Qbv shapes the traffic before entering the 802.11 MAC queues, preventing conflicting channel access from different queues. In particular, the scheduling enhancements of the IEEE 802.11ax (Wi-Fi 6) amendment are essential for supporting 802.1Qbv in wireless networks. By applying 802.1Qbv at the network stack, contention within the BSS is reduced. Additionally, multi-user Orthogonal Frequency Division Multiple Access (OFDMA) and triggered operations in 802.11ax improve channel access determinism.

B. OMNET++ AND INET

OMNeT++ [24] is a widely used discrete-event simulation framework designed for modeling various network types and communication systems. It is based on modular components, which serve as the building blocks for message exchange via connections and gates. The behavior of these modules is implemented in C++, while structural elements such as gates, connections, and parameters are described using the Network Description Language (NED). Initialization files (INI) define parameter values for model setup, supporting multiple values or ranges. By providing essential simulation functionalities, OMNeT++ enables users to focus on developing their simulation models.

The INET [25] framework extends OMNeT++ with a comprehensive library of network models, including communication protocols, network nodes, and connections. It also incorporates Internet stack models and both wired and wireless link layer protocols. This allows users to instantiate and integrate protocol layers, facilitating the design of custom hosts tailored to specific needs. Its modular architecture enables the rapid configuration of complex models, streamlining the simulation setup process.

III. RELATED WORK

Platooning has primarily been studied in vehicular networks, but its relevance is expanding to industrial scenarios, enabling efficient coordination of AGVs and cobots within smart factories [1]. However, in both contexts, the inherent mobility of devices introduces challenges, as packet loss and transmission delays can substantially impact the stability and performance of platoon operations [2], [6]. Furthermore, shared wireless communication channels necessitate effective medium access control (MAC) strategies. While CSMA/CA

has traditionally been employed to mitigate collisions, its performance in mobility-driven scenarios is often suboptimal, failing to provide the low-latency and high-reliability guarantees required for real-time coordination [26], [27].

Various solutions have been proposed in vehicular networking research to address the challenges mentioned above. Collisions avoidance side, Zang et al. [28] introduced a collision detection method in which transmitting nodes utilize full-duplex channel sensing to identify simultaneous transmissions. In cellular-based environments, researchers are working on improving scheduling algorithms to mitigate or detect packet collisions, particularly in scenarios where vehicles operate outside base station coverage (5G Sidelink) [16], [29], [30]. Shao et al. [31] proposed a MAC-layer protocol designed to handle the intermittent connectivity of vehicular networks. This protocol employs multichannel reservation and priority schemes to improve throughput and packet delivery ratio. Likewise, Jonsson et al. [32] introduced a polling-based MAC-layer protocol combined with a specialized transport layer retransmission mechanism to reduce message error rates in intra-platoon communications.

Other approaches assume that a designated platoon leader assigns time slots to platoon members, similar to Time Division Multiple Access (TDMA), defining specific transmission windows for communication [33], [34], [35]. For example, Fernandes and Nunes [33] developed a system that partitions the IEEE 802.11p control channel into slots to ensure stable intra-platoon communications. Along similar lines, Segata et al. [34] proposed a TDMA-based beacon dissemination strategy in which the platoon leader initiates beacon transmission, followed sequentially by other platoon members.

From a coordination perspective, several studies have explored algorithms for multi-lane platooning and reconfiguration maneuvers in response to environmental changes. For example, Firoozi et al. [36] investigated multi-lane platooning strategies that allow for adaptive reconfiguration. Deng et al. [37] proposed an optimization algorithm for coordinating the merging of autonomous vehicles at unsignalized intersections, ensuring safe and efficient maneuvering. Scholl et al. [38] developed a centralized platooning platform that optimizes truck platooning decisions based on predefined routes. More recently, Zhang et al. [39] analyzed scenarios where multiple vehicles attempt to join a platoon simultaneously. Their approach relies on the formation of local platoons that later merge into the main platoon, considering both vehicle position and speed to enhance merging efficiency and platoon stability.

In ultra-low latency and reliable wireless communications, several works have focused on integrating TSN with wireless protocols, e.g., 5G or Wi-Fi. In [40], the authors evaluate Wireless Time-Sensitive Networking (WTSN) capabilities, including time synchronization and time-based scheduling, over an IEEE 802.11-based network. The study, conducted in a collaborative robotic work cell with two robotic arms, analyzes application performance concerning network performance without considering mobility. In [41], the authors extend this evaluation to a mobile context, demonstrating how TSN redundancy reduces disruptions and delays caused by roaming and interference. In [42], the authors implement a 5G-based TSN system for cloud-controlled mobile robots. Their prototype integrates 5G with TSN, achieving end-to-end latency of less than 1 ms and 8 ms latency with 99.9% reliability in an industrial scenario.

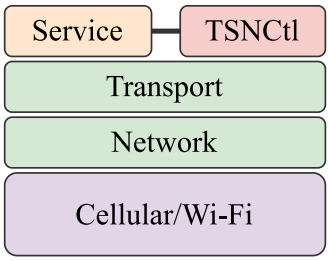


FIGURE 1. Integration of message and platoon controller in network stack.

Despite ongoing research efforts to improve the reliability of vehicular network communications, there is still a need for more technology-agnostic solutions that function independently of Wi-Fi or cellular technologies. Additionally, limited attention has been given to the platoon formation process [43], particularly in industrial settings involving mobile robots. Most existing research on platooning focuses on vehicle positioning to optimize energy consumption, lane utilization on highways, and intersection management. However, these works do not address the communication challenges inherent to dynamic platoon formation. In contrast, our approach is scenario-agnostic and prioritizes communication, ensuring efficient message dissemination and coordination among vehicles.

Our proposal introduces a TSN-like controller at the application layer that operates independently of the underlying wireless technology. This controller enables efficient platoon formation, including role definition (e.g., leader vs. follower), and improves packet delivery in intra-platoon communications by assigning each device a dedicated time slot for message dissemination. Unlike the approach in [43], our solution effectively manages platoon dynamicity, ensuring stability as devices join or leave the platoon. Moreover, by adopting a TSN-based scheme, our approach aligns with the increasing interest from both industry and academia in applying TSN principles to wireless environments.

IV. SYSTEM ARCHITECTURE

A. GENERAL OVERVIEW

The future of wireless communications is increasingly focused on exploiting TSN to ensure low latency, reliability, and determinism for critical services in wireless environments [18], [19], [27]. This work builds on the principle that smart factory devices can achieve synchronization through a centralized clock or by leveraging Radio and GPS technologies, enabling precise coordination and communication [44], [45], [46], [47]. Consequently, we do not

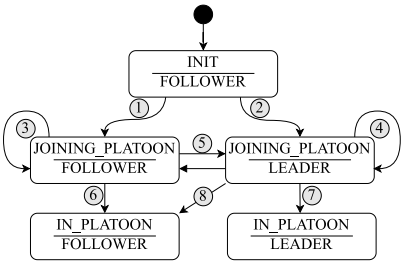


FIGURE 2. State diagram FSM TSNCtl.

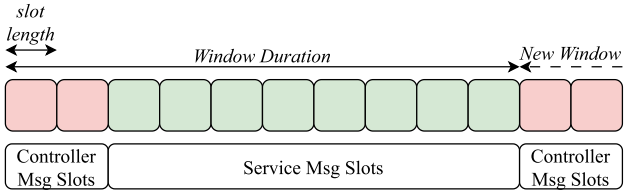


FIGURE 3. Slot-based scheduling.

focus on clock synchronization but concentrate exclusively on communication orchestration.

Our proposal integrates a TSN-like controller (TSNCtl) at the application layer of the networking stack (Fig. 1) within each device to manage message distribution during D2D communications. TSNCtl runs an FSM that facilitates platoon formation and conducts D2D communications using slot-based scheduling. The platoon formation process determines which devices will join the communication network, leveraging the *TSNCtl* FSM to initiate or facilitate platoon formation upon receiving service-related messages or detecting proximity to an existing platoon. During the platoon formation process, a platoon *leader* is elected from the devices intending to form a platoon, responsible for assigning communication slots to each member. These slots allow devices to communicate at specified times, reducing the chance of collisions during interactions with other vehicles in the platoon.

After the platoon is formed, *TSNCtl* intercepts messages created by the device's running service and determines the best timing for distributing them across the network based on the assigned slots. To manage this, *TSNCtl* maintains several queues, each with a unique priority level, where messages are stored before being sent out. Depending on the message type, *TSNCtl* allocates messages to the appropriate queue for prioritized delivery. Furthermore, *TSNCtl* tracks communication success within the network, removing nodes that fail to transmit during their assigned slots, reallocating those slots, and taking specific actions based on whether the departing node is a follower or the leader. This approach ensures reliability in highly dynamic environments, such as collaborative robots that need smooth coordination on a factory floor where network participants frequently change.

B. TSNCTL: PLATOON FORMATION AND SLOT ASSIGNMENT

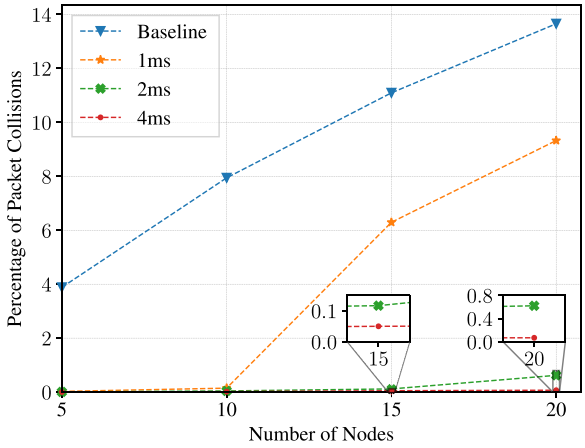
The *TSNCtl* FSM models the behaviors exhibited by devices during platoon formation. Fig. 2 illustrates the possible FSM states, which are defined by a combination of node state—*init*, *joining_platoon*, and *in_platoon*—and node role, either *follower* or *leader*, determining device responsibilities within the platoon.

During initialization, a device enters the *init* state, where key parameters such as communication window duration and slot length are set. The communication window defines the time interval for coordinating device activities within the platoon, while the slot length determines each transmission slot's duration, regulating message delivery. In this state, the internal *TSNCtl* component remains idle until the communication window elapses. As shown in Fig. 3, controller-based messages can only be exchanged at multiples of the window duration. On the other hand, the slot length determines the number of slots available per window; for instance, a 100 ms window with 10 ms slots results in 10 slots (Fig. 3).

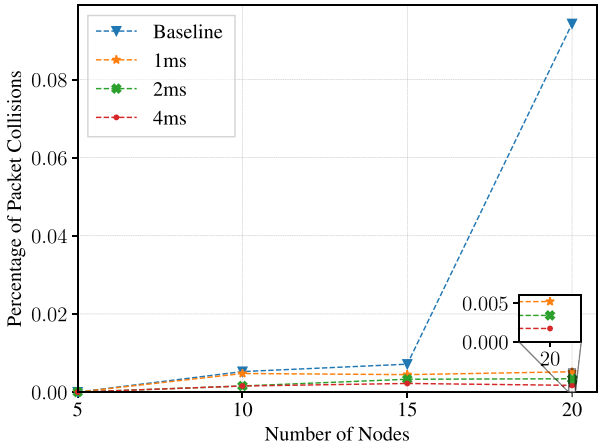
After the window elapses, the first two slots are reserved for control operations related to platoon formation and joining (red slots in Fig. 3). In the first control slot, nodes transmit controller messages containing node-specific information and a timestamp indicating message generation time. At this stage, nodes compete to become the *leader* of the platoon. To prevent collisions, controller messages are sent at randomly chosen times within the slot duration. In the second control slot, *TSNCtl* processes the messages received in the previous slot and changes its state in *joining_platoon*. As depicted in Fig. 2, in steps ① and ②, the node with the earliest timestamp—indicating the first generated controller message—becomes the *leader*, while all others assume the *follower* role. The *leader* then allocates communication slots to each *follower* based on policies that may consider factors such as the number of requested slots and the type of node (e.g., AGV, cobots, etc.).

If the *leader* node does not detect any neighboring nodes, it remains in its current state and reattempts platoon formation (step ④). A *leader* node that persists in this state may transit to *follower* role in the next controller message cycle (step ⑤). Conversely, if neighboring nodes are detected, it changes its state in *in_platoon* (step ⑦). Meanwhile, *follower* nodes await slot assignments from the *leader* and prepare to transmit data in their designated slots. Upon receiving the allocated slot, *follower* nodes confirm their role and transition to *in_platoon* (step ⑥). *Follower* nodes that do not receive a slot allocation remain in their state (step ③ in Fig. 2) and restart the platoon joining process in the next *TSNCtl* cycle.

In the *in_platoon* state, *follower* nodes transmit messages exclusively during their assigned slots, while the *leader* continues using the second control slot for communication. This mechanism also allows new nodes to integrate into the platoon. Once the platoon is formed, messages generated by the Service are classified based on priority and placed into designated queues within the controller. The *TSNCtl* then disseminates these messages according to their assigned slots and priority levels, ensuring efficient and organized communication within the platoon.



(a) Packet Collisions with a $1ms$ Request Interval



(b) Packet Collisions with a $1s$ Request Interval

FIGURE 4. Collisions in packet transmission with different platoon size.

C. TSNCTL: PLATOON REORGANIZATION

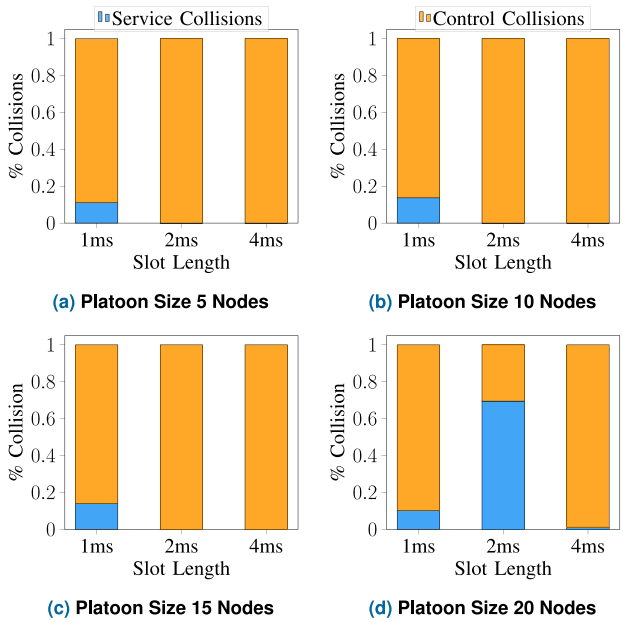
To maintain platoon stability in dynamic scenarios where nodes may enter or leave the platoon without prior notice, each *TSNCtl* instance tracks the nodes that successfully communicated during their assigned slots. If a node fails to transmit information during its allocated slot, it is removed from the platoon, and its slot is reallocated. This process involves distinct actions depending on whether the leaving node is a follower or the leader.

When a follower node exits the platoon, the leader node manages slot reallocation by compacting the slot schedule during the next available "*first*" controller slot. The leader shifts all subsequent allocations to fill the gap left by the departing node. In the following controller slot, the updated allocation is communicated to all remaining nodes in the platoon. This ensures the platoon's slot structure remains efficient and avoids protracted communication delays.

If the node exiting the platoon holds the leader role, different strategies are employed to ensure continuity. One option is to trigger a new platoon formation, resetting the state and initiating a new leader election (i.e., re-election method). Alternatively, the leader role can be directly transferred to the next node in the existing slot allocation (shift method). The latter strategy avoids the need for a new leader election, reducing message overhead and reorganization time. By maintaining the current platoon structure and simply transferring the leader role, the system ensures minimal disruption to ongoing operations.

It should be noted that nodes may leave the platoon after successfully transmitting a message during their allocated time slot. In such cases, their departure is only detected when they fail to participate in the following time window. This delayed detection necessitates careful consideration to avoid communication gaps and to ensure the efficient reallocation of resources (further details are provided in Section V).

The platoon reorganization procedure supported by *TSNCtl* relies on the shift method for leader reselection, ensuring robust and efficient platoon management, even in highly dynamic scenarios.



(a) Platoon Size 5 Nodes (b) Platoon Size 10 Nodes

(c) Platoon Size 15 Nodes (d) Platoon Size 20 Nodes

FIGURE 5. Impact of collisions by message type (1*ms*).

V. EVALUATION RESULTS

To demonstrate the usefulness and efficiency of our proposal, we developed the TSNCtl component outlined in Section IV within the OMNeT++ simulation library. We utilized INET 4.5 to simulate ad-hoc communications, specifically D2D scenarios, adhering to the IEEE-802.11p standard. Although IEEE 802.11p is traditionally used for V2V communications, it was chosen for this study as it effectively represents the Wi-Fi protocol commonly compared against the 5G sidelink, both applicable in industrial scenarios. Furthermore, we used the INET turtle mobility model to simulate device movement, where their speeds are generated through a uniform distribution, ranging from 2 to 5 m/s.

We implemented two OMNeT++ simple modules: a mock service (e.g., coordination service), sending messages to *TSNCtl* (Fig. 1), and a mock application, sending messages

directly on the socket. Both modules generate UDP packets with a regular interval of 100 *ms*. Packet sizes rely on the analysis conducted by the 5G Alliance for Connected Industries and Automation (5G-ACIA) [5] providing traffic models for different industrial use cases, including mobile robots and AGVs. Furthermore, we introduced a *spawner* module that generates and destroys devices within a specified area and frequency. For our testing, we fixed window duration to 100 *ms* and defined a 14,400-square-meter platoon formation area ($180m \times 80m$) corresponding to a typical factory [5].

In this environment, we initially examined the behavior of our model by varying the number of devices within a platoon and their entry frequency. Next, we assessed the resilience of our solution under different packet sizes. Finally, we evaluated the robustness of the proposed controller by altering the frequency at which devices were generated and removed, analyzing three distinct scenarios: low, medium, and high generation/removal frequencies.

These experiments were performed on a Linux VM running OMNeT++ having 16 CPUs and 32 GB of RAM. Each experiment was repeated ten times to ensure statistical reliability. It should be noted that, at this stage, the experiments do not consider signal attenuation due to obstacles like objects and walls within a factory.

A. ASSESSING PACKET COLLISIONS BY VARYING PLATOON SIZES

The first set of experiments involves investigating the performance of our proposed solution (Section IV) when varying the number of nodes to be included in the platoon. These experiments assume a maximum packet size of 256 bytes [5], a window duration of 100 ms, 1 minute simulation time, and three slot lengths: 1, 2, and 4 ms. Fig. 4 compares the percentage of packet collisions between scenarios using CSMA/CA (*Baseline*) and those utilizing the *TSNCtl* controller introduced in this work.

Fig. 4(a) shows the percentage of packet collisions obtained when nodes attempt to form a platoon with an interval of 1 ms between their request, representing a highly synchronized scenario. This could occur, for example, when mobile robots in a smart factory simultaneously receive commands to execute a specific action, such as asking devices to move materials or goods across the factory. Conversely, Fig. 4(b) presents results for a scenario where nodes initiate platoon formation with a 1-second interval, simulating a more generalized and less synchronized use case where nearby nodes join the platoon over time to optimize resource utilization.

The *TSNCtl* controller effectively mitigates packet collisions and supports efficient platoon formation in both scenarios. In Fig. 4(a), the discrepancy between *TSNCtl* and the *Baseline* case is particularly marked, as the percentage of collisions does not exceed 1% in the TSNCtl scenarios using 2 and 4 ms slot lengths compared to an average of 9.14% in *Baseline* cases. Conversely, in Fig. 4(b), the gap between

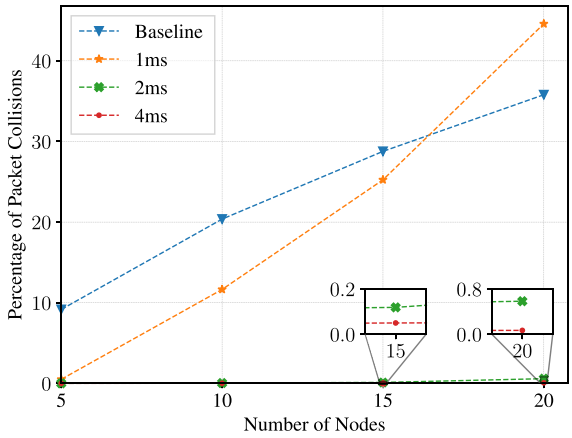


FIGURE 6. TSNCtl in a vehicular use case.

TSNCtl and the *Baseline* is less pronounced, as the average collision rate remains below 1% even in the *Baseline* cases. This suggests the system's sensitivity to packet collisions naturally decreases with more randomized communication intervals.

It should be noted, however, that our proposal does not perform well with slot lengths that are too small. As Fig. 4 demonstrates, with 1 ms slot length, the collisions start growing with more than 10 nodes, meaning that *TSNCtl* struggles with forming the platoon. Hence, to exploit the full potential of our proposal, selecting an appropriate slot length is imperative. To further analyze this aspect, Fig. 5 provides a zoom-in on each message type's impact on the total collisions generated (service or control), considering the high dynamic scenario (interval of 1 ms). A lower blue bar indicates better algorithm performance, implying that most collisions are confined to control messages. The results reveal that a 1 ms slot length is unreliable, as it leads to collisions in service-related packets, compromising the reliability of the solution. The same happens in 2 ms slot length but only in scenarios with over 20 nodes.

Additionally, increasing packet sizes exacerbates performance degradation. For instance, in a V2V use case based on the CAR2CAR Communication Consortium's analysis [48], where the maximum packet size is set to 800 bytes, slot lengths below 2 ms show significantly reduced performance. The generated collisions could prevent the correct formation of the platoon (Fig. 6). In contrast, with larger slot lengths (e.g., 4 *ms*), the *TSNCtl* consistently achieves near-zero collision rates, ensuring robust and reliable platoon management.

In industrial scenarios, where high reliability and strict determinism are paramount, even minimal packet collisions can disrupt operations or cause delays in task execution. By consistently maintaining near-zero collision rates, *TSNCtl* ensures the predictability and reliability necessary for time-sensitive applications, such as coordinating autonomous robots or optimizing resource allocation in smart factories.

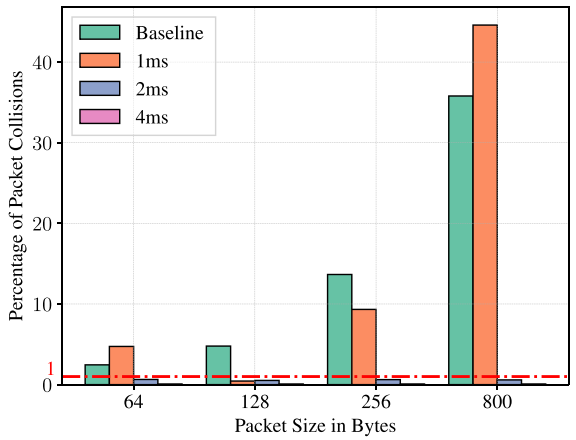


FIGURE 7. Collisions in packet transmission with different packet size.

B. ASSESSING PACKET COLLISIONS BY VARYING PACKET SIZES

In this set of experiments, we further analyze the behavior of our solution when varying the packet size in 1-minute simulations. We consider a platoon of 20 vehicles generated under the previously defined settings at intervals of 1 ms. Fig. 7 illustrates the percentage of packet collisions observed when varying packet sizes across different slot lengths. The packet dimensions analyzed are derived from the data provided in [5] for industrial scenarios and [48] for vehicular scenarios.

The results show a clear trend for both *Baseline* and *TSNCtl*, where the percentage of collisions increases as packet size grows. As mentioned earlier, using a 1 ms slot length proves unreliable, displaying unexpected behaviors. For example, with 64 bytes packets, the percentage of collisions is unexpectedly higher than with 128 bytes packets. Similarly, packet sizes above 256 bytes exhibit a significant increase in collisions, highlighting the limitations of this configuration.

Fig. 7 also demonstrates the increasing trend that *Baseline* has when increasing the size of the packet transmitted. It shows an absolute deviation of around 11% from the average. In the vehicular scenario (800 bytes), it reaches over 35% collisions while remaining below 15% for the industrial scenario. Conversely, *TSNCtl* shows a consistent collision rate, staying below 1% for slot lengths greater than 1 ms. Indeed, the absolute deviation from the average collision rate is approximately 3.9% for a 2 ms slot length and just 0.8% for a 4 ms slot length.

These results validate that *TSNCtl* effectively mitigates collisions when an appropriate slot length is chosen, maintaining packet collisions below 1% in both industrial scenarios (64–256 bytes) and vehicular scenarios (>256 bytes). This also highlights the robustness and adaptability of *TSNCtl* in addressing the challenges of dynamic communication environments.

TABLE 1. Dynamics study.

Dynamics Level	Detection Time Mean	Detection Time Max	Collision Percentage
LOW	107.045 ms	172.839 ms	1.4×10^{-5}
MEDIUM	107.111 ms	192.439 ms	1.6×10^{-5}
HIGH	105.393 ms	185.048 ms	3.3×10^{-5}

C. TSNCTL ROBUSTNESS IN HIGHLY DYNAMIC ENVIRONMENTS

Finally, we assess the robustness of the platoon after its formation by varying the proportion of nodes affected by join or leave events within a given time window. Three scenarios were designed to represent different levels of dynamics: *low*, *medium*, and *high*. In the *low* dynamics scenario, nodes join every 2 seconds, with a maximum of two nodes leaving per minute. The *medium* dynamics scenario features 3 to 5 nodes leaving per minute, while the *high* dynamics scenario involves 4 to 12 nodes leaving per minute. The simulation time is 10 minutes, slot length is 4 ms, and other settings maintain the same mobility and area parameters as those used in other experimental setups.

Table 1 presents the average time required to detect when a node leaves the platoon. We have not included the time needed to reorganize the platoon, as this depends on the slot length. Indeed, once a departure is detected, the new slot allocation is communicated during the controller slot immediately following the detection. Specifically, detection and slot reallocation are completed within the "*first*" controller slot, and the communication of this update occurs in the next controller slot. Consequently, as shown in the table, the average detection time is approximately one time window (around 107 ms). This value can increase when nodes leave the platoon immediately after communicating in their allocated time slot, as described in Section IV. In such cases, the maximum detection can be close to two time windows; in the simulated scenarios, we obtained 183.44 ms on average (Detection Time Max).

It is relevant to highlight that detection time is entirely independent of the level of network dynamics. The primary impact of increased dynamicity is reflected in the percentage of collisions, which remains negligible, consistently below 0.5%. This ensures the system maintains high reliability and determinism in platoon management even under high dynamic conditions.

VI. DISCUSSION AND FUTURE WORKS

The approach proposed in this work aligns with the increasing interest from academia and industry in leveraging TSN for wireless message propagation [19], [49]. This trend has been further accelerated by the introduction of the latest IEEE 802.11 standard–IEEE 802.11be, also known as Wi-Fi 7 [23]–which aims to fully integrate TSN functionalities to support ultra-reliable, low-latency communications.

5G Integration. Beyond its immediate advantages, our approach introduces new research opportunities and has the potential to reshape message dissemination within platoons. As demonstrated in this work, a TSN-enabled D2D environment can significantly enhance packet delivery ratios and overall network performance in large-scale industrial and vehicular deployments. Additionally, integrating TSN with cellular networks presents a promising path for supporting time-sensitive and low-latency applications [3], particularly within 5G infrastructures [50]. This integration could also yield substantial benefits for D2D communications utilizing cellular technologies.

Platoon Formation. Future research will focus on extending our solution to more realistic environments, taking into account signal attenuation due to obstacles and device trajectories, while also developing intelligent error-handling strategies for platoon formation. Furthermore, devices located at the boundary of two platoons must be managed effectively and could even serve as relays, facilitating inter-platoon message forwarding. This would extend the communication range and enable seamless integration between adjacent platoons.

Synchronization. Our current implementation relies on a shared time reference, achieved through GPS, radio synchronization, or a centralized clock within an industrial setting. However, time synchronization introduces additional technical challenges in more complex scenarios where not all devices are inherently synchronized. To address this, the next phase of our solution will incorporate a wireless implementation of the PTP protocol, enabling devices to first establish a synchronized time domain before initiating communication.

Messages Prioritization. In dynamic communication scenarios, devices may need to exchange messages with varying priority levels. Our solution can leverage a TSN-based approach to implement differentiated message queues, particularly in delay-sensitive applications. By distinguishing between critical and non-critical messages, network resources can be allocated more efficiently, ensuring timely delivery of high-priority information and improving overall system reliability. Additionally, priority-based queues could enhance the rapid and dependable dissemination of time-sensitive messages across multiple platoons, supporting stable operations in dynamic environments.

VII. CONCLUSION

This study presents an innovative approach to enhancing D2D communications and facilitating platoon formation through a TSN-like, slot-based scheduling solution, marking a first step toward deterministic communication in direct communication networks. Our TSNCtl controller effectively orchestrates message dissemination by reducing packet collisions and ensuring reliable data exchange between mobile nodes. Performance results, measured within the OMNeT++ simulation environment, demonstrate the effectiveness of our approach, particularly in scenarios with varying platoon sizes and packet sizes. We also assessed the robustness of the proposed controller by varying the events generated within intra-platoon communications.

In the future, we plan to enhance the TSNCtl component to support multi-platoon communications, leverage sensor data for improved platoon formation, and integrate priority-based message queuing. Additionally, we will investigate the impact of varying network performance parameters and assess the feasibility of deploying the controller without prior synchronization.

REFERENCES

[1] A. Bozzi, S. Graffione, J.-F. Jimenez, R. Sacile, and E. Zero, "A platoon-based approach for agv scheduling and trajectory planning in fully automated production systems," *IEEE Trans. Ind. Informat.*, vol. 21, no. 1, pp. 594–603, Jan. 2025.

[2] M. A. Javed, F. U. Muram, S. Punnekkat, and H. Hansson, "Safe and secure platooning of automated guided vehicles in industry 4.0," *J. Syst. Archit.*, vol. 121, Dec. 2021, Art. no. 102309. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1383762121002125

[3] "Using 5G sidelink in industrial factory applications," 5G-ACIA, Kuwait City, Kuwait, White Papers, 2023, Accessed: Jan. 10, 2025. [Online]. Available: https://5g-acia.org/whitepapers/using-5g-sidelink-in-industrial-factory-applications/

[4] J. Axelsson, "Safety in vehicle platooning: A systematic literature review," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1033–1045, May 2017.

[5] "A 5G traffic model for industrial use cases," 5G-ACIA, Kuwait City, Kuwait, White Papers, 2019, Accessed: Jan. 10, 2025. [Online]. Available: https://5g-acia.org/whitepapers/a-5g-traffic-model-for-industrial-use-cases/

[6] 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., 2015.

[7] S. Zeadally, J. Guerrero, and J. Contreras, "A tutorial survey on vehicle-to-vehicle communications," *Telecommun. Syst.*, vol. 73, pp. 469–489, Mar. 2020.

[8] *IEEE Standard for Information Technology–Local and Metropolitan Area Networks–Specific Requirements—Part-11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments*, IEEE Standard 802.11p-2010, 2010.

[9] "Intelligent transport systems (ITS); ITS-G5 access layer specification for intelligent transport systems operating in the 5 Ghz frequency band," ETSI, Sophia Antipolis, France, Rep. 302 663 V1.3.1, Oct. 2019.

[10] *IEEE Standard for Wireless Access in Vehicular Environments (WAVE)–Networking Services*, IEEE Standard 1609.3-2020, 2016.

[11] "Technical specification group services and system aspects; release 14 description; summary of Rel-14 work items; (Release 14), Version 14.0.0," 3GPP, Sophia Antipolis, France, Rep. TR 21.914, 2018.

[12] J. Meredith, F. Firmin, and M. Pope, "Release 16 description; summary of Rel-16 work items," 3GPP, Sophia Antipolis, France, Rep. TR 21.916, 2022.

[13] R. Cupek et al., "Autonomous guided vehicles for smart industries–the state-of-the-art and research challenges," in *Proc. 20th Int. Conf. Comput. Sci. (ICCS)*, Amsterdam, The Netherlands, 2020, pp. 330–343.

[14] H. Andreasson et al., "Autonomous transport vehicles: Where we are and what is missing," *IEEE Robot. Autom. Mag.*, vol. 22, no. 1, pp. 64–75, Mar. 2015.

[15] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The future of industrial communication: Automation networks in the era of the Internet of Things and industry 4.0," *IEEE Ind. Electron. Mag.*, vol. 11, no. 1, pp. 17–27, Mar. 2017.

[16] Y. Jeon, S. Kuk, and H. Kim, "Reducing message collisions in sensing-based semi-persistent scheduling (SPS) by using reselection lookaheads in cellular V2X," *Sensors*, vol. 18, no. 12, p. 4388, 2018. [Online]. Available: https://www.mdpi.com/1424-8220/18/12/4388

[17] A. Nasrallah et al., "Ultra-low latency (ULL) networks: The IEEE TSN and IETF DetNet standards and related 5G ULL research," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 88–145, 1st Quart., 2018.
[18] "Wireless TSN: Tests and use case demonstrations with Wi-Fi," AVnu Alliance, Beaverton OR, USA, White Paper, Feb. 2024, Accessed: Jan. 28, 2025. [Online]. Available: https://avnu.org/resource/wireless-tsn-tests-and-use-case-demonstrations-with-wi-fi-white-paper/
[19] D. Cavalcanti, C. Cordeiro, M. Smith, and A. Regev, "WiFi TSN: Enabling deterministic wireless connectivity over 802.11," *IEEE Commun. Stand. Mag.*, vol. 6, no. 4, pp. 22–29, Dec. 2022.
[20] *IEEE Standard for Local and Metropolitan Area Networks–Timing and Synchronization for Time-Sensitive Applications*, IEEE Standard 802.1AS-2020 (Revision of IEEE Std 802.1AS-2011), 2020.
[21] A. Garbugli, A. Bujari, and P. Bellavista, "End-to-end QoS management in self-configuring TSN networks," in *Proc. 17th IEEE Int. Conf. Factory Commun. Syst. (WFCS)*, 2021, pp. 131–134.
[22] Y. Peng, B. Shi, T. Jiang, X. Tu, D. Xu, and K. Hua, "A survey on in-vehicle time-sensitive networking," *IEEE Internet Things J.*, vol. 10, no. 16, pp. 14375–14396, Aug. 2023.
[23] T. Adame, M. Carrascosa-Zamacois, and B. Bellalta, "Time-sensitive networking in IEEE 802.11 be: On the way to low-latency WiFi 7," *Sensors*, vol. 21, no. 15, p. 4954, 2021.
[24] OMNeTpp. "Omnet++ event discrete simulator." 2024. Accessed: Jan. 7, 2025. [Online]. Available: https://omnetpp.org/
[25] "INET framework." Accessed: Jan. 7, 2025. [Online]. Available: https://inet.omnetpp.org/
[26] R. Reddy G and R. R, "An empirical study on MAC layer in IEEE 802.11p/Wave based vehicular ad hoc networks," *Procedia Comput. Sci.*, vol. 143, pp. 720–727, Nov. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050918321410
[27] "Wireless time-sensitive networking (TSN)," AVnu Alliance, Beaverton OR, USA, White Paper, 2020, Accessed: Jan. 28, 2025. [Online]. Available: https://avnu.org/wp-content/uploads/2014/05/Avnu-WirelessTSN-white-paper-V1.0_Final.pdf
[28] J. Zang, V. Towhidlou, and M. Shikh-Bahaei, "Collision avoidance in V2X communication networks," in *Proc. IEEE Wireless Commun. Netw. Conf. Workshop (WCNCW)*, 2019, pp. 1–6.
[29] T. Maruko, S. Yasukawa, R. Kudo, S. Nagata, and M. Iwamura, "Packet collision reduction scheme for LTE V2X sidelink communications," in *Proc. IEEE 88th Veh. Technol. Conf. (VTC)*, 2018, pp. 1–5.
[30] L. Cao, S. Roy, and H. Yin, "Resource allocation in 5G platoon communication: Modeling, analysis and optimization," *IEEE Trans. Veh. Technol.*, vol. 72, no. 4, pp. 5035–5048, Apr. 2023.
[31] C. Shao, S. Leng, Y. Zhang, A. Vinel, and M. Jonsson, "Performance analysis of connectivity probability and connectivity-aware MAC protocol design for platoon-based VANETs," *IEEE Trans. Veh. Technol.*, vol. 64, no. 12, pp. 5596–5609, Dec. 2015.
[32] M. Jonsson, K. Kunert, and A. Böhm, "Increased communication reliability for delay-sensitive platooning applications on top of IEEE 802.11 p," in *Proc. 5th Int. Workshop Commun. Technol. Veh.*, 2013, pp. 121–135.
[33] P. Fernandes and U. Nunes, "Platooning with IVC-enabled autonomous vehicles: Strategies to mitigate communication delays, improve safety and traffic flow," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 1, pp. 91–106, Mar. 2012.
[34] M. Segata et al., "Towards inter-vehicle communication strategies for platooning support," in *Proc. 7th Int. Workshop Commun. Technol. Veh.*, 2014, pp. 1–6.
[35] Y. Park and H. Kim, "Collision control of periodic safety messages with strict messaging frequency requirements," *IEEE Trans. Veh. Technol.*, vol. 62, no. 2, pp. 843–852, Feb. 2013.
[36] R. Firoozi, X. Zhang, and F. Borrelli, "Formation and reconfiguration of tight multi-lane platoons," *Control Eng. Pract.*, vol. 108, Mar. 2021, Art. no. 104714. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0967066120302847
[37] Z. Deng, K. Yang, W. Shen, and Y. Shi, "Cooperative platoon formation of connected and autonomous vehicles: Toward efficient merging coordination at unsignalized intersections," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 5, pp. 5625–5639, May 2023.
[38] J. Scholl, N. Boysen, and A. Scholl, "E-platooning: Optimizing platoon formation for long-haul transportation with electric commercial vehicles," *Eur. J. Oper. Res.*, vol. 304, no. 2, pp. 525–542, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221722003113
[39] M. Zhang, C. Wang, W. Zhao, J. Liu, and Z. Zhang, "A multi-vehicle self-organized cooperative control strategy for platoon formation in connected environment," *IEEE Trans. Intell. Transp. Syst.*, vol. 26, no. 3, pp. 4002–4018, Mar. 2025.
[40] S. Sudhakaran, K. Montgomery, M. Kashef, D. Cavalcanti, and R. Candell, "Wireless time sensitive networking impact on an industrial collaborative robotic workcell," *IEEE Trans. Ind. Informat.*, vol. 18, no. 10, pp. 7351–7360, Oct. 2022.
[41] S. Sudhakaran et al., "Zero-delay roaming for mobile robots enabled by wireless TSN redundancy," in *Proc. IEEE 19th Int. Conf. Factory Commun. Syst. (WFCS)*, 2023, pp. 1–8.
[42] P. Kehl et al., "Prototype of 5g integrated with TSN for edge-controlled mobile robotics," *Electronics*, vol. 11, no. 11, p. 1666, 2022.
[43] L. Ghiro, R. L. Cigno, E. Tonini, S. Fontana, and M. Segata, "Can platoons form on their own?" in *Proc. IEEE Veh. Netw. Conf. (VNC)*, 2024, pp. 180–187.
[44] H. Baniabdelghany, R. Obermaisser, and A. Khalifeh, "Extended synchronization protocol based on IEEE802.1AS for improved precision in dynamic and asymmetric TSN hybrid networks," in *Proc. 9th Mediterr. Conf. Embed. Comput. (MECO)*, 2020, pp. 1–8.
[45] K. Alemdar, D. Varshney, S. Mohanti, U. Muncuk, and K. Chowdhury, "RFClock: timing, phase and frequency synchronization for distributed wireless networks," in *Proc. 27th Annu. Int. Conf. Mobile Comput. Netw.*, New York, NY, USA, 2021, pp. 15–27. [Online]. Available: https://doi.org/10.1145/3447993.3448623
[46] A. Garg, A. Yadav, A. Sikora, and A. S. Sairam, "Wireless precision time protocol," *IEEE Commun. Lett.*, vol. 22, no. 4, pp. 812–815, Apr. 2018.
[47] K. F. Hasan, Y. Feng, and Y.-C. Tian, "GNSS time synchronization in vehicular ad-hoc networks: Benefits and feasibility," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 12, pp. 3915–3924, Dec. 2018.
[48] V. Martinez and F. Berens, "Survey on ITS-G5 cam statistics," CAR 2 CAR Commun. Consortium, Braunschweig Germany, Rep. C2CCC_TR_2052, Dec. 2018. [Online]. Available: https://www.car-2-car.org/fileadmin/documents/General_Documents/C2CCC_TR_2052_Survey_on_CAM_statistics.pdf
[49] M.-T. Thi et al., "IEEE 802.1 TSN time synchronization over Wi-Fi and 5G mobile networks," in *Proc. IEEE 96th Veh. Technol. Conf. (VTC)*, 2022, pp. 1–7.
[50] Z. Satka, M. Ashjaei, H. Fotouhi, M. Daneshtalab, M. Sjödin, and S. Mubeen, "A comprehensive systematic review of integration of time sensitive networking and 5G communication," *J. Syst. Archit.*, vol. 138, May 2023, Art. no. 102852.

ANGELO FERAUDO (Graduate Student Member, IEEE) received the M.Sc. degree in computer engineering from the University of Bologna, Italy, in 2020. He is currently pursuing the Ph.D. degree in computer science and engineering with the University of Bologna, where he collaborates with the Mobile Middleware Research Group. During his Ph.D., he visited the Department of Computer Science, University of York, U.K., and the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA, USA, where he was involved in O-RAN research. His research interests include vehicular networks, next-generation networks (5G/6G), O-RAN, and vehicular computing integration in cloud-continuum spectrum.

ANDREA GARBUGLI (Member, IEEE) received the M.Sc. degree (with Hons.) in computer science engineering from the University of Bologna, Italy, where he is currently a Postdoctoral Researcher. His research focuses on ultra-low latency communications and network acceleration in the edge cloud.

PAOLO BELLAVISTA (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in computer science engineering from the University of Bologna, Italy, where he is currently a Full Professor of Distributed and Mobile Systems. He was recently the Scientific Coordinator of the H2020 BigData Project IoTwins and is the representative of the University of Bologna within the Industry4.0 Competence Center called BI-REX. His research activities span from pervasive wireless computing to online big data processing under quality constraints, from cloud continuum middleware to digital twins for Industry 4.0 applications. He serves on several editorial boards, including IEEE COMMUNICATIONS SURVEYS AND TUTORIALS (Associate Editor-in-Chief), *ACM Computing Surveys*, *Journal of Network and Computer Applications* (Elsevier), and *Pervasive and Mobile Computing*.