IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 30, 2021, accepted October 13, 2021, date of publication October 15, 2021, date of current version October 26, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3120769

Timely Survey of Time-Sensitive Networking: Past and Future Directions

YOUHWAN SEOL, DOYEON HYEON, JUNHONG MIN, MOONBEOM KIM, AND JEONGYEUP PAEK, (Senior Member, IEEE)
Department of Computer Science and Engineering, Chung-Ang University, Seoul 06974, Republic of Korea

Corresponding author: Jeongyeup Paek (jpaek@cau.ac.kr)

This work was supported in part by the National Research Foundation of Korea (NRF) funded by the Korea Government (MSIT) under Grant NRF2021R1A2C100884011, and in part by the Chung-Ang University Graduate Research Scholarship in 2020.

ABSTRACT Time-sensitive networking (TSN) is a next generation local area network technology for the coexistence of information and operation technology, targeted to industrial automation, in-vehicle networks, and avionic networks in the industrial Internet of Things. This paper presents an up-to-date survey of the research efforts in TSN over the past few years, including the latest efforts in standardization by the IEEE. We review more than 170 TSN-related academic research papers published by major academic publishers and categorize these studies according to the topic, purpose, and methodology, analyzing research trends in TSN. The *TSN core function* deals with the functional and internal perspective to achieve ultra-low latency/jitter and ultra-reliability. *Network management* includes work with macroscopic and external views, such as configuration, management, and maintenance of TSN-enabled networks. In particular, we analyze the characteristics of existing studies that have evaluated the TSN network and propose future directions to advance the limits of TSN for real-world deployment. We aim this study to be a cornerstone for fellow and new researchers.

INDEX TERMS Time-sensitive networking (TSN), industrial network, real-time Ethernet, local area network (LAN), the Internet of Things (IoT).

I. INTRODUCTION

The Ethernet [1] is a powerful technology capable of meeting the communication needs of vast number of general purpose applications, and has become commonplace in the Internet of Things (IoT). Traffic from IP-based applications such as web browsing, audio/video streaming, or sensing data from IoT devices are essentially best-effort, and supported on the Ethernet or Wi-Fi, which share the same service philosophy.

Ethernet technology has proved to be quite successful, although no real-time support for the IEEE standard Ethernet existed until recently. Real-time communication is an essential requirement in some industries; thus, it has been developed for specific tasks or domains through numerous proprietary protocols, such as EtherCAT, PROFINET, and Sercos III for industrial automation; controller area network (CAN) and FlexRay for automotive; and avionics full-duplex switched Ethernet (AFDX) for avionics. These protocols perform their designated tasks capably but have compatibility limitations for converging Ethernet-like networks and devices.

As each solution is tailored to a specific application, expanding of these solutions to other industries is also limited. Therefore, the IEEE Time-Sensitive Networking (TSN) [2] task group has been vigorously active since 2012, standardizing real-time capabilities in the Ethernet.

TSN describes several mechanisms for ensuring or improving the real-time transmission of Ethernet traffic and defines the first IEEE standard for time-triggered message delivery in switched Ethernet networks [3].[1] The key feature of the TSN standard is to allow flows with various requirements, such as ultra-low latency, low-jitter, and zero-loss reliability, to be controlled on an integrated bridged network. To achieve deterministic real-time communication over the Ethernet for tightly demanding applications, TSN uses globally synchronous time and schedules that route messages across multiple network components. By defining a queue that transmits messages according to a time schedule, TSN

The associate editor coordinating the review of this manuscript and approving it for publication was Lei Shu.

[1] The TSN standard does consider several Layer 2 technologies [3], but the focus is still mainly on the Ethernet [1].

guarantees deterministic bounded latency for time-critical reserved traffic, integrating all traffic classes and applications into a single standard-based Ethernet network. In addition to standardization's documentation and administration perspective, the technological and practical perspectives of maximizing efficiency through flexible prioritization and fine-tailored transmission emphasize the importance of TSN at the core of information technology (IT) such as Industry 4.0, cyber-physical system, and industrial Internet of Things (IIoT).

This paper is an up-to-date statistical survey of research on TSN (as of the end of 2020), aiming to be an informative source to those who wish to conduct academic research on TSN. Instead of rewriting a tutorial [4]–[6], we present information and pointers that we wished we had known when we first entered the field of TSN. Our work is motivated by the fact that many early studies on TSN are outdated, as TSN was officially standardized only recently (2018 [3]) and is still an ongoing and evolving effort. Some research has even been innocently incorrect because it was based on discussions and terms from prestandard drafts.

Specifically, when we research new technology, we ask the following questions:

- What is it, and what are the key challenges?
- What topics are researchers working on? What has been focused on so far, and what is expected in the future?
- What methodologies are used for this research?
- What applications, scenarios, and topologies are considered?
- What kind of devices or simulators are used for experiments or simulations, and what is the scale?

The goal is to provide up-to-date answers to these questions to aid and promote new researchers in TSN.

The remainder of this article is structured as follows. First, in Section II, we clarify the scope, methodology, and taxonomy of this work. Section III provides a statistical analysis of the surveyed papers to discover the temporal and demographic research trends and investigate the methodology, setup, and scale used for TSN evaluations. Based on the classification, Sections IV, V, and VI briefly review the relevant studies in each subtopic, and Section VII summarizes what has been learned and the critical future challenges in TSN. Lastly, Section VIII concludes this study.

II. OVERVIEW

We first define the scope of the survey and the taxonomy in this work. A note follows this discussion regarding the related tutorials for TSN.

A. SCOPE OF THE SURVEY

This paper limits the survey scope to TSN-based studies to the extent possible and does not consider studies on audio/video bridging (AVB) [7] or time-triggered Ethernet (TTEthernet) [8] for the following reasons. First, studies on AVB have sufficient results for time-sensitive traffic with predictable latency (looser than TSN, but tighter than the best-effort), and have been well covered as a predecessor to TSN. For those who want to learn more about AVB, Teener *et al.* provided details in [9]. Second, most studies on the TTEthernet have also been passed to TSN or conducted concurrently because time-triggered traffic scheduling is similar to gate control list (GCL) scheduling of TSN. Zhao *et al.* elaborately compared both in [10].

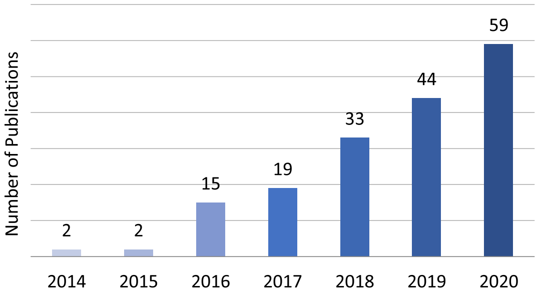


FIGURE 1. Number of surveyed publications per year.

This decision excludes remarkable studies on extensions to Ethernet-based real-time communication protocols, such as AFDX or flexible time-triggered Ethernet (FTT-Ethernet), among others. However, we believe this decision provides a more accurate view of the work done on TSN because the above protocols are generally expected to be integrated into IEEE TSN as standardization progresses.

We searched academic research publications on *IEEE Xplore*,[2] *ACM Digital Library*,[3] and *Google Scholar*[4] with the keywords "TSN", "Time-Sensitive Network" and "Time-Sensitive Networking" from 2014 to 2020. Additionally, we went through the references, following papers that cited or were cited by the papers, but studies not found in these databases were excluded. Among these, we reviewed 174 research publications.

The numbers of publications collected for each year are plotted in Fig. 1. After the AVB Task Group changed its name to TSN Task Group, research papers related to the TSN have been published since 2014, and the work volume has steadily increased. As of 2017, 38 research papers were published. Then, the number has increased rapidly since 2018 when IEEE 802.1Q-2018 [3] with TSN key features was officially standardized reaching 59 publications in 2020. Fig. 2 illustrates an overview of TSN standardization history. In addition to the quantitative increase in TSN research, a steady effort has been made to standardize TSN. We believe that this growing trend will continue because the ultra-low latency/jitter or multiclass traffic converged LAN/MAN is expected to become a critical foundation technology. This technology will be required in various industries in the future, such as industrial automation and in-vehicle networks, which are the current main targets. In addition, from the perspective of 5G,

[2] https://ieeexplore.ieee.org
[3] https://dl.acm.org
[4] https://scholar.google.com

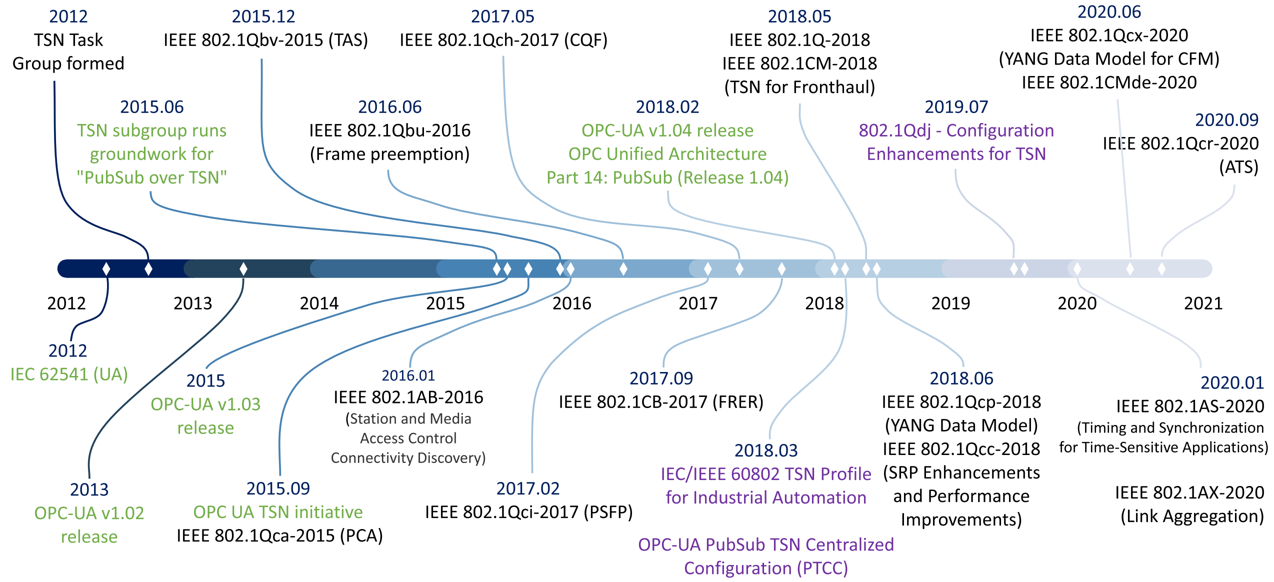


FIGURE 2. History of IEEE TSN standardization (with OPC-UA).

which is one of the megatrends of the current technology development, TSN is of great importance as a 5G fronthaul to support traffic classes transmitted and multiplexed through the packet-based switching fabric. Further, as TSN still has technical challenges in standardization and implementation issues and fundamental challenges that must be addressed or evolved, more research publications are expected over the next few years.

B. TAXONOMY

We believe that the most insightful way to classify previous work is to categorize it by *purpose* (i.e., What are the primary challenges behind the work?). Based on this, the taxonomy was divided into the following three main categories with 12 subtopics.

1) TSN CORE FUNCTION

In this category, we report work in which the subject is TSN internal functions. TSN is a toolbox providing various functions, not a single technology or protocol. Several functions can be selected individually or unitively according to the requirements of the target application that need to enable TSN characteristics. Therefore, this category is divided again into several subtopics based on individual functionality.

- *Time-Aware Shaper (TAS):* IEEE 802.1Qbv *Enhancements to Traffic Scheduling* [11] pursues the zero-jitter transmission of critical traffic through open/close scheduling of the GCL in each switch's egress port. We present work that analyzes the performance of the TAS (primarily compared with other shapers) or enhances the TAS and provides insight on the scheduling synthesis approach (SSA). Although the TAS is a novel feature for ultra-low-latency transmission of time-critical traffic, it has a high implementation complexity and additional overhead due to the GCL schedule generation/deployment and time synchronization. Moreover, it is unsuitable for aperiodic traffic flows. Accordingly, TSN also provides other shapers suitable for the requirements of other traffic types.
- *Credit-Based Shaper (CBS):* For more relaxed latency-bound traffic, IEEE 802.1Qav *Forwarding and Queueing of Time-Sensitive Streams* [12] specifies two traffic classes: AVB-A (less than 2 ms per seven hops) and AVB-B (less than 50 ms per seven hops), and defines the CBS that is relatively simple to implement compared to the TAS. The CBS allocates a logical bandwidth for each class by regulating a 'credit' to limit the bursts and prevent the starvation of lower priority classes. The characteristics of the CBS can be regarded as a token-bucket-based per-class shaper.
- *Asynchronous Traffic Shaper (ATS):* IEEE 802.1Qcr [13] defines ATS, which achieves, as the name suggests, limited low-delay transmission without global time-synchronization. The ATS is flexible in handling mixed traffic, including aperiodic traffic, and does not require time synchronization unlike the TAS.
- *Frame Preemption (FP):* The basic concept of IEEE 802.1Qbu *Frame Preemption* [14] is that *express* traffic can interrupt *preemptable* traffic. In detail, when express traffic is ready to be transmitted on a given switch egress port, even if a (preemptable) frame is in the state of being transmitted, the transmission halts immediately, and the transmission of express traffic starts. When the express traffic transmission is completed, the transmission of preempted traffic is resumed.

Further, IEEE 802.3br *Interspersing Express Traffic* [15] abstracts from the PHY layer to provide preemptable MAC and express MAC service interfaces with additional MAC merge-sublayer in egress ports.

- *Stream Reliability:* As networks that consider TSN primarily aim for deterministic transmission of critical traffic (e.g., safety-related), one of the key differences in comparing AVB with TSN is reliability. Several industrial reliability architectures exist, such as high availability and seamless redundancy (HSR) [16] and parallel redundancy protocol (PRP) [17]. Nevertheless, TSN published a stand-alone standard IEEE 802.1CB *Frame Replication and Elimination for Reliability (FRER)* [18], which facilitates reliability via redundancy by applying the stream concept.

2) NETWORK MANAGEMENT

In this category, TSN research work from a macroscopic viewpoint, such as network component discovery, negotiation of management or configuration information (e.g., GCL schedule deployment), or user interaction, is organized. These studies range from the simplest single TSN domain to integrating TSN with legacy real-time or wireless communication systems.

- *Time Synchronization:* Because most TSN standards require a global time reference, time synchronization between network components is one of the most critical issues in TSN. Moreover, IEEE 802.1AS *Timing and Synchronization for Time Sensitive Applications* [19] defines the generic precision time protocol (gPTP) as a new profile that adapts IEEE 1588-2008 precision time protocol (PTP) [20] for TSN environments. The precision and efficiency of time synchronization techniques directly affect TSN performance; thus, several studies have been conducted to improve the mechanisms in IEEE 802.1AS.
- *Framework/Modeling:* The anticipated quality-of-service (QoS) can be achieved only when complex and sophisticated TSN functionalities work as intended. Thus, each network component must know the relevant configuration and information promptly. In this subtopic, with the background of IEEE 802.1Qcc *Enhancements to Stream Reservation Protocol* [21] standard and centralized management, we introduce studies that present network models to communicate configurations among network components efficiently. We also report studies on fancy frameworks for network designers that facilitate network analysis, configuration, or monitoring.
- *Case Study/Proof-of-Concept:* As TSN is considered for applications in various target domains, designing a specialized network configuration for a specific purpose or requirement is common. Therefore, in this subtopic, studies that analyze how to configure TSN components efficiently and studies that serve as a cornerstone for replacing or integrating with the legacy network are covered.
- *Wireless Integration:* The advantages of wireless, such as the convenience of physical deployment and mobility, are essential elements for industrial automation. Ultimately, extending representative wireless communication technologies, such as 5G or Wi-Fi, to TSN is essential to construct a wired/wireless integrated real-time industrial network. However, as TSN is primarily an Ethernet-based wired system, various additional studies (e.g., clock synchronization, protocol conversion, and radio link failure handling) are underway for incorporation into wireless environments.

3) FOR TSN RESEARCHERS

The previous two categories summarize studies for the internal function and overall use of TSN. In this category, we organize information for researchers beginning TSN research to grasp where to begin quickly, for example, which simulators and real devices support TSN features or which stage of research for these features is underway. We also introduce important but inevitably unclassified studies, such as introductory documents or reports on the industry status for TSN.

- *Simulator:* A network simulator can significantly reduce costs when designing and validating new protocols or identifying new scenarios. We introduce TSN simulators based on proven simulator frameworks, such as OMNeT++ [22] and Riverbed [23], or studies that implement specific libraries or modules based on those frameworks.
- *Hardware:* Representative commercial off-the-shelf (COTS) equipment, or even development or evaluation kits for academic research, are lacking in TSN. Nevertheless, some papers have investigated practical challenges of TSN support in COTS equipment, or proposed hardware designs to support various functions of TSN more harmoniously while directly implementing in application-specific integrated circuit (ASIC), field programmable gate array (FPGA) or systems-on-a chip (SoC).
- *Miscellaneous:* Through tutorials for beginners or reports on the future applicability of TSN technology, it is possible to obtain a rough understanding of why the technology is necessary, what features and characteristics it has, and how it operates internally or with the surroundings. Interested readers can gain preparatory knowledge through these documents before studying TSN in earnest.

In classifying TSN studies of various subjects, it is sometimes challenging to separate studies from each other in an entirely disjointed manner, especially those dealing with combined or converged subjects with ambiguous classification boundaries (e.g., a performance comparison between multiple shapers). However, as one of the significant contributions of this study is tracking changes in research trends,

TABLE 1. Number of time-sensitive networking related research papers in each subtopic, tabulated by publication year.

Category	Subtopic	2014	2015	2016	2017	2018	2019	2020	Total	Sim.	Exp.
TSN Core Function	Time-Aware Shaper	0	1	6	5	8	9	17	**46**	6	3
	Credit-Based Shaper	0	1	1	2	1	3	1	**9**	5	0
	Asynchronous Traffic Shaper	0	0	1	1	3	3	3	**11**	7	0
	Frame Preemption	0	0	1	1	0	1	1	**4**	3*	2*
	Stream Reliability	1	0	0	1	4	4	3	**13**	1	0
		1	**2**	**9**	**10**	**16**	**20**	**25**	**83**	**22**	**5**
Network Management	Time Synchronization	0	0	0	0	1	0	4	**5**	3	1
	Framework/Modeling	0	0	5	3	3	10	8	**29**	3	11
	Case Study/Proof-of-Concept	0	0	0	3	1	2	7	**13**	4	8
	Wireless Integration	0	0	0	1	2	5	9	**17**	5	5
		0	**0**	**5**	**7**	**7**	**17**	**28**	**64**	**15**	**25**
For TSN Researchers	Simulator	0	0	1	2	3	4	2	**12**	12*	1*
	Hardware	1	0	0	0	1	2	2	**6**	0	4
	Miscellaneous	0	0	0	0	6	1	2	**9**	0	0
		1	**0**	**1**	**2**	**10**	**7**	**6**	**27**	**12**	**5**
	Total	**2**	**2**	**15**	**19**	**33**	**44**	**59**	**174**	**49 (28.2%)**	**35 (20.1%)**

'*' indicates using both evaluation methods.

we aggregate the granular classification criteria to eliminate duplicate counting in the statistical analysis section. In contrast, the review section includes as many references as necessary even if they appear more than once, to reduce the misunderstanding of research results during this aggregation process.

C. RELATED TUTORIALS ON TSN

To focus this paper on a survey of the current state-of-the-art TSN studies and their statistical analyses, we decided not to deal with tutorials on TSN in this paper. This decision was because we wanted to focus on providing concise pointers to interested readers and because a few fine tutorials have already been published [4]–[6], [24]. We aim to provide information that is not in those tutorials.

For example, Nasrallah *et al.* [4] regarded IEEE TSN and the IETF deterministic network (DetNET) as L2 and L3, respectively, of an ultra-low-latency network, and provided a comprehensive survey and broad overview on both standards. However, the subjects of the papers they surveyed are weak in the distinction between the TSN and AVB (or TTEthernet) research heritage, and the time range of the papers they surveyed ends in 2018, after which TSN was standardized [3]. Therefore, it is necessary to update many novel studies (over 100) conducted since then.

Bello *et al.* [24] focused more on TSN and provided a relatively detailed overview of the standards and mechanisms related to the transmission scheme for time-critical traffic, which is the core of TSN. It also provides an analysis of two primary TSN use cases, industrial communication and automation system, as a reference for TSN integration or applicability in both domains. However, they discuss other domains or the general-purpose use of TSN less.

These tutorials are all good references for an overview of TSN and understanding the basic mechanisms of TSN core functions; therefore, we recommend them. However, to the best of our knowledge, our work is the most up-to-date (as of the end of 2020) and the first to include a statistical analysis of research trends and topic categorization in TSN. We believe this paper provides valuable information to readers who want to conduct academic research on TSN.

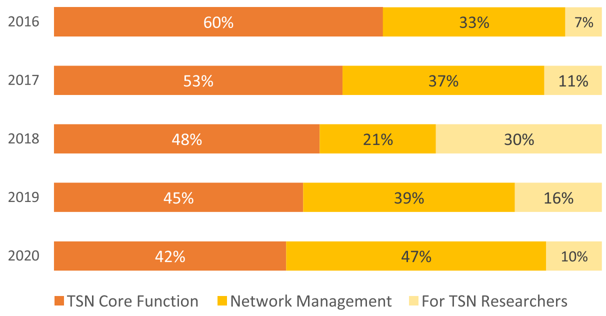


FIGURE 3. Changes in research trends.

III. STATISTICAL ANALYSIS

This section presents the statistical analysis of TSN studies gathered according to the method discussed in Section II-A. Research papers are summarized and analyzed based on the subtopic classification, evaluation methods, publication outlet, and demographics.

A. SUBTOPIC CLASSIFICATION

Table 1 lists the subtopics of the three main categories and publication statistics for each subtopic by year. First, the *TSN core function* category was researched the most with 83 publications. As expected, TAS has been studied the most with 46 papers of the five subtopics of the *TSN core function* category.

Early TAS research focused on implementing a solution for *flow-based* Qbv in a multihop full-switched TSN network while being inspired by the existing *frame-based* scheduling of TTEthernet. Based on this, more drastic and sophisticated SSAs were proposed to solve more complex problems, including various additional components (e.g., mixed-criticality or joint routing). In addition, studies for TAS enhancements and studies that analyze TAS performance in various environments are included.

Second is *framework/modeling*, covered by 29 papers, which ranked no.1 in 2019. Starting with the research on using TAS efficiently in 5G fronthaul [25] in 2017, research on *wireless integration* has increased rapidly, reaching 17 papers. Next, *stream reliability* and *case study/proof-of-concept* have 13 publications each.

Fig. 3 illustrates the relative publication ratio of three categories from 2016 to 2020. Most notably, despite the growing number of publications on the *TSN core function*, their overall share of TSN research has been decreasing. Instead, the sum of the other two categories has been steadily increasing in publication count. Especially in 2020, the *network management* category recorded its largest number of publications, with more than the *TSN core function* for the first time. This trend means the efforts on *how to implement the basic functions of the TSN* have been paying off, leading to focusing on *how to use TSN well*.

B. EVALUATION METHOD

Proper evaluation is essential to make the tremendous effort devoted to standardizing and enhancing TSN productive. The two rightmost columns of Table 1 display the statistics of the evaluation methods: simulation or experiment.

Overall, 49 studies performed simulations, and 35 studies conducted experiments with real devices. Two studies used both methods (marked with *). Thus, a total of 82 studies evaluated their work using either simulation or experiments, which is less than half of all surveyed papers (47.12%). Fortunately, as TSN research matures, the work ratio evaluated through simulations or experiments has also increased. Using simulations or experiments, 15 and 8 studies (23 of 44; 52.27%) in 2019 and 18 and 21 studies (39 of 59; 66.10%) in 2020 evaluated TSN (or the proposed enhancement), respectively.

For the most popular subtopic, TAS, few papers have conducted simulations or experiments because most TAS studies (primarily related to SSA) have employed numerical analysis methods calculating the proposed objective functions and constraints using a solver (e.g., satisfiability modulo theories (SMT) or integer linear programming (ILP)) or network calculus. Another reason is that decent TSN simulators that support TAS (e.g., NeSTiNg [26]) have only emerged since 2018, and later to support on real devices. Only three of the TAS studies have conducted experiments with real devices, among which two of these papers have evaluated their proposed SSA only in simulations and merely deployed their presynthesized schedules to devices for experiments. Likewise, *stream reliability* had only one study conducting simulations. In contrast, CBS, ATS, and FP have relatively lower implementation complexity, and more papers have conducted simulations or experiments.

It is disappointing that cutting-edge proposals for TAS improvements have not performed simulation- or experiment-based evaluations due to the lack of an appropriate environment. However, this situation does not mean that a basic performance evaluation for TAS has not been conducted. Some studies on *framework/modeling* and *case study/proof-of-concept* have implemented a relatively simple network topology with a basic TAS, and evaluated the proposed architecture through simulations or experiments.

C. NETWORK SCALE

We counted the nodes and time-critical flows used in evaluations to demonstrate how closely TSN studies mimic the current real-world scenarios. We were motivated to do this for the following reasons. As many studies have already revealed, the time required for GCL scheduling is dependent on the network size and number of scheduled traffic (ST) flows [27]–[31]. For TSN to be deployed in the real world, it must complete GCL scheduling within a reasonable time for a generally applicable network size and number of flows. This criterion is generous in that it does not consider the worst cases where network topology is dynamic.

The following guidelines have been established for a high-level understanding of the network scale used for simulations and experiments in vast number of papers with diverse evaluation setups and methods. First, all TSN-enabled switches (SWs) or bridges (including logical bridges) in the evaluation are counted as SWs, and all other network devices other than SWs (i.e., talker, listener, background traffic generator, traffic sink, etc.) are counted as end systems (ESs). Similarly, we count all scheduled flow/stream/traffic as ST flow. Although this may exclude the specificity of each paper, it allows us to understand the overall trend of the TSN evaluation scale.

Further, we targeted all scenarios in which the motivation for each experiment is distinguished, but in the case of a single experiment with multiple network topologies, only the largest-scale topology was selected to focus on the scalability. We believe that by selecting the most daring scenarios attempted by each paper, we demonstrate the limits of the network scale used by TSN evaluations.

Referring to the experimental settings of each paper with the guidelines above, we count the number of SWs, ESs and ST flows used in the evaluation of TSN studies conducted so far and obtained some basic statistics. Table 2 presents the number of scenarios used in simulation- and experiment-based studies and the median, average, and maximum SWs, ESs, and ST flows in those scenarios.

In total, 44 scenarios were used for simulations, and the network topology of each scenario had an average of 7.5 SWs and 12.91 ESs. Among them, 37 scenarios simulated the TAS with an average of 8.89 ST flows. The simulation scenario with the most SWs was by Gutiérrez *et al.* [32], who analyzed the time synchronization accuracy of a large network

TABLE 2. Network scale in simulation/experiment-based evaluations for time-sensitive networking research.

Eval.	Elem.	Scene.#	Med.	Avg.	Max.
Sim.	SW	44	3	7.50	100
	ES	44	8	12.91	103
	ST	37	5	8.89	50
Exp.	SW	24	2	2.63	7
	ES	24	3	4.17	14
	ST	21	1	2.52	13

using 100 time-aware nodes with up to 100 hops to the grandmaster clock. Moreover, Nsaibi *et al.* [33] presented a simulation scenario involving the largest number of ESs to analyze the feasibility of integrating TSN into a legacy real-time Ethernet for automation systems. Further, Hellmanns *et al.* constructed a line-topology consisting of 50 SWs, 52 ESs, and 50 ST-flows. They compared the performance of stream-based scheduling, class-based scheduling, and FP [34]. This study used most SWs in a simulation-based evaluation that included a performance analysis of TAS scheduling.

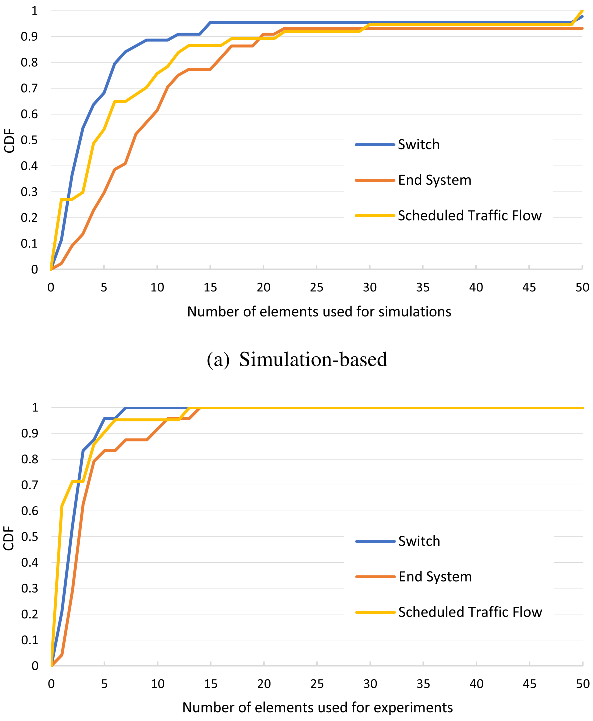
For real-device experiments, fewer scenarios were used than simulation-based evaluations. The network scale was also smaller with an average of 2.63 SWs, 4.17s ES, and 2.52 ST flows. Nevertheless, Vlk *et al.* validated the applicability of GCL schedules with two TSN SWs (capability stack: 802.1AS, 802.1Qbv, 802.1Qcc, NETCONF, and OPC-UA Pub/Sub), 14 ESs (13 actuators and one PLC emulated by the IXIA traffic generator), and 13 ST flows [35]. The scenario using the largest number of SWs is a study comparing the performance of TAS and FP in a multi-hop TSN network (maximum of six hops) using seven SWs (capability stacks: 802.1AS, 802.1Qbv, and 802.1Qbu) [36].

To understand the distribution of the network scale used in simulation- and experiment-based evaluations, we plotted the CDF in Fig. 4. Of the studies involving simulation, 80% of the studies used only six or fewer SWs. The number of SWs used in the experiments was even smaller. More than half of the scenarios used only two SWs, and 83% of the scenarios included three or fewer SWs.

These numbers are disappointing. More evaluation must be done on larger topology to demonstrate the satisfiability of TSN's target specifications, such as the worst-case delay of 100 μs over five hops for ST. In addition, we point out that, while some studies that evaluated the work using numerical analyses have actively used a case study from real-life scenarios, such as the Orion crew exploration vehicle (CEV) [37] or General Motors [38], they have not been simulated. The first study has 31 ESs, 15 SWs, and 100 ST flows, and the second has 20 ESs, 20 SWs, and 27 ST flows, so they are a meaningful experiment target scenario as they represent an appropriate network size for an actual use case.

D. SIMULATOR FRAMEWORKS

We investigated the types of simulators used in 49 studies that conducted simulations. The use frequency of the simulators is presented in Fig. 5, referring to the dominant frameworks and their characteristics. This information can be helpful to TSN researchers aiming to include simulation-based evaluation in their work.



(a) Simulation-based

(b) Experiment-based

FIGURE 4. Cumulative distribution function (CDF) of network scale (number of switches, end systems, and scheduled traffic flows) used to evaluate TSN research.

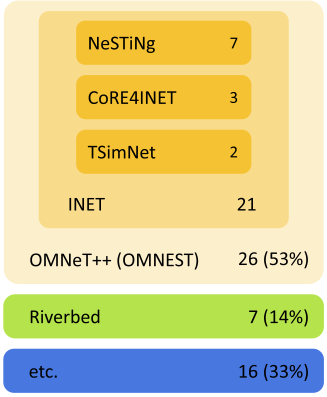


FIGURE 5. Statistics for the simulator framework usage.

First, OMNeT++/OMNEST has been used in 26 papers (23 and three respectively), more than half of the studies that include simulation. Moreover, INET, which is often used together, was used in 21 publications.[5] OMNeT++ [22] is an

[5] When counting simulators, if only the extension frameworks are indicated in the paper, we assumed that the corresponding base framework was also used. For example, in papers that only stated NeSTiNg, both OMNeT++ and INET were also counted.

TABLE 3. Current state of affairs of real devices and supported time-sensitive networking standards.

Id.	Type	Model	IEEE 802.1								Ref.
			AS	Qbv	Qbu	Qci	Qcc	Qav	CB	Qat	
1	Analog Device RAPID-TSNEK-v0001	Development kit	O	O	O	O	O		O		[40]–[42]
2	TTTech Edge IP Solution EVM		O	O	O	O	O		O		[36], [43], [44]
3	NXP LS1021A TSN		O	O		O		O			
4	SoCe MTSN kit		O	O	O	O	O	O	O	O	[45]
5	IPMS IP-Core for TSN		O	O	O	O	O	O	O		
6	Kontron KBOX C-102-2 TSN		O	O	O		O				
7	Hilscher netX 90 Starter-kit		O	O	O						
8	Cisco IE-4000	Off-the-shelf	O	O							[46], [47]
9	InnoRoute TrustNode		O	O		O					[48]–[51]
10	NXP SJA1105 Ethernet Switch		O	O		O		O			[52]
11	Kontron PCIe-0400-TSN	PCIe NIC	O	O	O		O				
12	HP Intel I210		O			O		O		O	[40], [41], [53]–[55]
13	Marvell 88Q6113	Processor chip	O	O				O		O	
14	Broadcom BCM53570		O	O	O	O	O		O		

extensible, component-based C++ simulation framework for building a network simulator, and OMNEST is a commercial version that provides more features, and is compatible with OMNeT++. OMNeT++ is a simulation framework without models. Thus, it is primarily used in combination with external frameworks and libraries, such as INET [39], containing models for the Internet TCP/IP stack or wired/wireless link layer protocols. In addition to providing various features, both OMNeT++ and INET are open source based on the GPL license; allowing researchers to use thees when implementing a TSN simulator.

NeSTiNg, CoRE4INET, and TSimNet are representative extension frameworks that implement specialties for TSN and can be used appropriately as needed for TSN simulation. A novel simulation framework especially designed for TSN, NeSTiNg includes 802.1Qbv, Qbu, Qav, and VLAN tagging, and has been used in seven papers, which is the highest frequency among the three despite being the newest. Next, CoRE4INET, originally an extension for simulating real-time Ethernet such as TTEthernet, can simulate AVB and TSN using 802.1Qbv, Qav, and Qci extensions and has been used in three papers. Finally, TSimNet, which primarily implements non timed-based features, such as 802.1Qbu, Qci, and CB, was used in two papers.

The Riverbed simulator has gradually expanded its capabilities through various studies and has been used in seven papers so far. Similarly, TSN functional modules based on the Riverbed simulator have been implemented by several researchers. In addition, some studies use other simulators, such as ns-3, UPPAAL, ALEVIN, PAnSim, and Mininet, which we categorized as 'etc.,' along with seven papers that did not indicate what simulator was used.

E. EXAMPLES OF REAL-WORLD TSN DEVICES

We cannot sufficiently emphasize the need for better equipment and a development environment for researchers to study fast-evolving TSN. From this perspective, we investigated various products for reference to establish TSN now or in the near future. Table 3 lists the names of products by major vendors and the TSN standards currently supported by each product. This list is only an example of TSN-supported devices, not a complete list. All the information is based on the official webpages or flyers that introduce each product. In addition, the rightmost column lists papers that explicitly indicate the use of a related product, and Fig. 6 shows the images of some example products. Several development kits, including SoC and the TSN solutions, offer relatively many features and a high degree of freedom in implementation. In contrast, off-the-shelf SWs allow the ease of configuration and use, although support for additional functionality may be relatively limited and slow. In addition, PCIe NIC can be used to support TSN functionality in ESs.

F. PUBLICATION DEMOGRAPHICS AND OUTLETS

Fig. 7(a) plots publication distribution of demographics based on the first author of each paper. Germany overwhelmingly ranked first in terms of the number of publications, at 60, more than a third of all publications, even more than Asia and

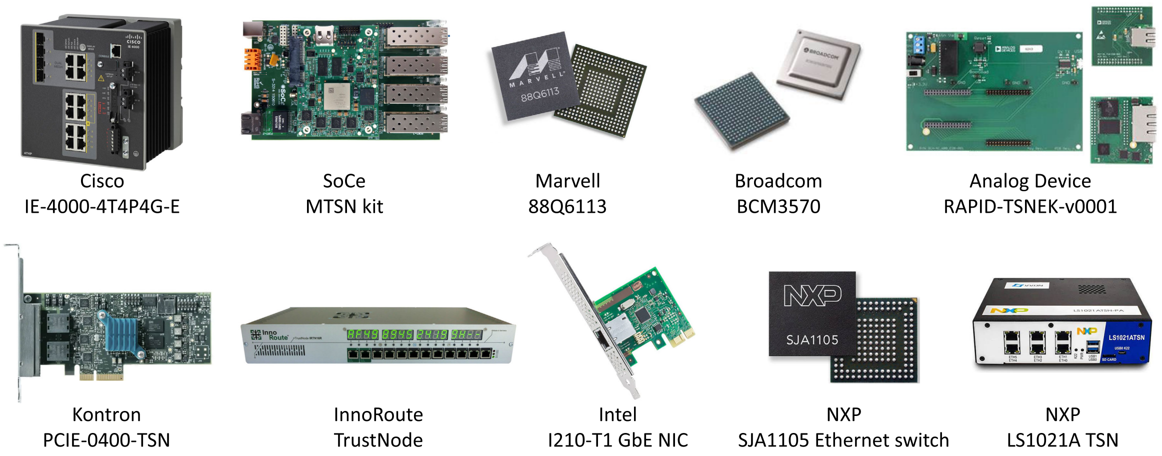
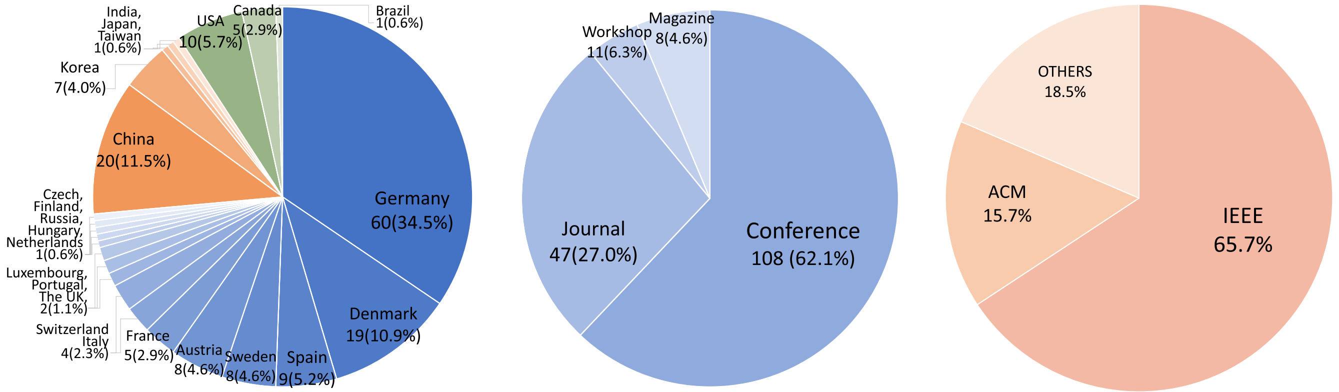


FIGURE 6. **Examples of TSN-supported real devices.**



(a) Demographics (b) Publishing outlet (c) Organization

FIGURE 7. **Distribution of publication information.**

America combined. As Germany launched Industry 4.0 as a national strategic initiative, these overwhelming statistics are believed to result from their enthusiastic research on ICT and IoT for the digitization of manufacturing and industrial practices. Denmark (19 papers), Spain (nine papers), Sweden (eight papers), and Austria (eight papers) contributed to a combined total of 73.56% of research from Europe. In Asia, China is ranked second overall with 20 papers, and the Republic of Korea (seven papers) and Japan, India, and Taiwan (one publication each) followed. North and South America published papers from the United States (10 papers), Canada (five papers), and Brazil (one paper). Out of the 24 countries in Fig. 7(a), 15 new participants are listed in TSN research after 2018, showing that countries are increasingly interested in TSN. We believe that this trend will continue in the coming years, leading to innovative research in more regions.

Fig. 7(b) plots the distribution of outlet types in TSN-related publications. Overall, 27% of the papers were published in journals, 55.8% were published in conference proceedings, 6.3% papers were published in symposiums or workshops, and 4.6% papers were published in magazines. In particular, 13 papers were published in the *IEEE Access* journal, and 26 papers were published by the *IEEE Emerging Technologies and Factory Automation* conference, the two most popular publishing outlets.

As illustrated in Fig. 7(c), 65.7% of all papers were published in IEEE-sponsored publishing outlets, 15.7% were in ACM-sponsored outlets, and 18.5% were published by other publishers, such as Elsevier, Springer, and MDPI, which is unsurprising considering that TSN is an IEEE standard.

IV. TSN CORE FUNCTIONS

The most crucial aspect of a time-critical system is to ensure that timing requirements are always met. The *TSN core functions* are primarily significant enhancements to IEEE 802.1Q in the form of amendments, and most have already been quickly integrated into the IEEE 802.1Q-2018 standard [3]. According to the required QoS, various traffic types are classified by the name of each stream (or flow) and are given a definite priority. Through prescheduled queue gate control [11] or exclusive resource reservation [12], TSN ensures QoS requirements of high-priority streams. Furthermore, transmission integrity is also as crucial. For example, imagine having a necessary appointment. Finding the way to arrive there on time is just as important as scheduling time for it. In addition, it is also necessary to have a backup plan for unexpected catastrophes.

This section presents studies on *TSN core functions* for ensuring ultra-low latency and reliability for time-critical traffic in multihop switched networks. For readers who require a more detailed understanding of the introduced features, please refer to the representative tutorial papers recommended in each subsection.

A. TIME-AWARE SHAPER

Frame-level scheduling is an NP-complete problem. The general formulation using SMT or ILP was carried over from TTEthernet research or even earlier. This carryover is because the method that enables the transmission of hard real-time traffic based on perfectly synchronized time is similar to the GCL scheduling in TAS that schedules the open-close times of the egress port [56]–[60]. Although this is a somewhat valid approach, it has limitations [27]. Oliver *et al.* articulated these limitations and presented a gate window-level approach based on the first-order theory of arrays, which is close to the principle method of GCL scheduling. Reusch *et al.* [61] extended this window-level synthesis approach to improve flexibility by mitigating mutual isolation constraints between ST flows.

Early SSAs did not consider the possibility of additional routing paths (i.e., simply using the shortest-path bridging that is the most basic use of off-the-shelf SWs). This fixed-route scheduling is the most intuitive to select a fast route by minimizing the number of hops but has latent problems, such as load imbalance and frequent bottlenecks. These problems occur because many flows, including low-priority flows, select the same path. Moreover, because scheduling on a fixed route is formulated in a relatively small solution space, it is not sufficiently representative of the performance, and schedules with better performance that are feasible can be excluded. Altogether, the latent disadvantage of assigning multiple time-critical flows to a small number of specific paths becomes more pronounced as the overall network load increases.

Consequently, research is underway to determine optimized scheduling in the expanded solution space by considering various paths of flows (i.e., solve routing and scheduling problems jointly). However, it is exceedingly difficult to determine an optimal solution within a limited amount of time because routing constraints must be considered in addition to scheduling constraints that are already highly time-complex. Therefore, methods are proposed to obtain practically satisfactory solutions using meta-heuristics, such as the greedy randomized adaptive search procedure, genetic algorithm, or Tabu search [28], [29], [62]–[73]. Using a completely different approach, such as source routing [74], is also proposed.

Although the primary purpose of SSA is generating a schedule for the deterministic transmission of time-critical traffic, due to the nature of TSN, where various traffic types coexist, the transmissions of lower priority traffic must also be considered. This consideration can be achieved with the proper configuration of scheduled queues, according to the IEEE 802.1Qbv standard, and with routing according to traffic class [75]–[80].

In this mixed-criticality environment, a *guard band* (GB) can be a useful option. It can be used to ensure the exact start time of the ST transmission by preventing low priority traffic (e.g., best effort) from interfering with the ST transmission. However, GB is a time interval in which no traffic can be transmitted; thus, the bandwidth is wasted. Therefore, there are SSAs that reduce the size or number of GBs in GCL [81], [82].

Another option for increasing the efficiency of TAS is IEEE 802.1Qch *Cyclic Queuing and Forwarding (CQF)* [83]. This CQF allows the bridge to periodically synchronize the frame enqueue/dequeue operations with bounded latencies that depend only on the number of hops and cycle time. The goal is to reduce the dependency on the network topology and increases determinism. It can also be used in combination with *ingress policing* from IEEE 802.1Qci *Per-Stream Filtering and Policy* [84] to support more accurate guarantees of an allotted cycle time in TAS, especially for isochronous traffic with high priority [85], [86].

While proposals for novel SSAs with higher scalability and schedulability continue, various analysis methods have also been continuously proposed to assess the performance of these SSAs better [34], [38], [53], [87]–[97]. In particular, Bello *et al.* [95] conducted a comprehensive simulation-based analysis that includes all of TSN's major shaping features, such as TAS, CBS, FP, and GB, and the corresponding traffic types such as Class ST, Class A, and Class B.

B. CREDIT-BASED SHAPER

Many notable studies on IEEE 802.1Qav and CBS exist from the AVB era, inherited by TSN. Nevertheless, because TSN needs to define a separate traffic class for ST traffic so that they are guaranteed to be transmitted in isolation from other traffic types while the highest traffic class number has already been occupied by AVB,[6] CBS enhancements considering this were continuously studied [98]–[102]. Similarly, analysis of latency and backlog bounds of AVB traffic in TSN has actively been researched [103]–[106].

C. ASYNCHRONOUS TRAFFIC SHAPER

The ATS originated from the urgency-based scheduler (UBS) [107], which implements a per-flow interleaved regulator [108] based on rate-controlled service disciplines, providing deterministic latency with low implementation

[6] To be precise, 'traffic class' and 'priority' numbers are two different, but related terms in the IEEE 802.1Q standard. The recommended mapping from 'priority' to 'traffic class' depends on the number of queues supported by the switch, and is configurable.

complexity. Specht and Samii [107] analyzed the key parameters of UBS and presented a UBS parameter synthesis approach based on the topology rank solver heuristic and SMT [109]. Through these works, the authors address the shortcomings of traditional asynchronous schedulers to provide deterministic low-latency. Le Boudec [108] also proved that interleaved traffic regulators, which are the key concept behind the ATS, do not increase the worst-case latency. The IEEE 802.1Qcr standard provides a paternoster scheduling (PS) solution.

Many studies have analyzed the performance of ATS using various methods, such as numerical analysis and simulations [108], [110]–[114]. Zouh *et al.* provided process modeling for both solutions (UBS and PS) and analyzed the differences [115], [116]. In addition, the authors presented the first hardware implementation using FPGAs [117]. Prados-Grazon *et al.* used reinforcement learning to cope with the complexity of asynchronous based flow scheduling [118]. Nevertheless, there is no work until 2020 that includes a formal formulation of the flow scheduling/configuration of asynchronous TSN networks, as in synchronous ones. Samii and Specht [107] used STM to solve the problem, but their work does not include an explicit formulation to express the constraints of the problem. This could be a future research direction for asynchronous TSN.

D. FRAME PREEMPTION

Frame preemption allows selected express traffic to move faster with relatively low implementation and management complexity (no time synchronization required). When the requirements are less fastidious, it can be used solely by itself to guarantee the bounded worst-case end-to-end latency of express traffic [36]. Moreover, FP allows reducing the GB size when used with TAS [119], [120].

Zhao *et al.* used network calculus to analyze the theoretical worst-case delay upper bound of the influence on AVB traffic by time-triggered traffic with FP [105]. In addition, the influence of FP has been evaluated using various methods, such as numerical analyses [121], simulations [34], [95], [119], [120], or experiments [36], [121]. In particular, Gogolev and Bauer [36] implemented 802.1AS, TAS, and FP in SoC to compose a simple and practical network with six-hop daisy-chained bridges and nonsynchronized end devices for real experiments. Their analysis indicates that FP can demonstrate relatively reasonable performance in the transmission of high-priority traffic in networks where TAS is unsuitable, but it does not consider the effects of FP on low-priority traffic.

E. STREAM RELIABILITY

Kehrer *et al.* provided a FRER overview and compared the differences with other redundancy protocols [122]. The spatial redundancy concept of FRER is robust against both permanent defects, such as physical link failure or temporary defects, such as frame drops. Spatial redundancy is a method to ensure that there is no problem in data transmission, even if a failure occurs in a specific location, by sending duplicate packets through multiple paths. The TSN standard separately defined maximally redundant tree multipath algorithm for this purpose. However, this does not allow the user or main controller to set the route statically; thus, the algorithm for setting the static route has also been studied by several researchers.

Despite the intention of FRER to eliminate duplicate frames seamlessly, the network resource consumption for indiscriminately redundant transmissions is an obvious drawback. Therefore, appropriate mixing with temporal redundancy to minimize unnecessary transmissions in a so-called proactive manner has been studied [123], [124].

In terms of cost efficiency, critical traffic is the main target of guaranteeing reliability. Therefore, fault tolerance of ST can be considered in the GCL scheduling stage [30], [125]–[130]. Atallah *et al.* proposed a degree-of-conflict aware iterative routing and scheduling approach [30], which is a greedy heuristic that creates a schedule to minimize the degree-of-conflict between generated graphs. The authors calculate the degree of conflict by analyzing conflict elements among streams and creating a graph by grouping streams with high interdependence. A disjointed routing set that shares the same source and destination is created using this process, providing a traffic multi-redundancy level. The authors also proposed a fault-resilient joint topology routing and schedule synthesis that provides tolerance levels of critical traffic by generating multiple disjoint paths like [30] in their study [127].

Well-scheduled transmissions can prevent errors, such as frame drop due to interference from other traffic (lower or same priority). However, streams with stringent requirements also consider the redundancy level when scheduled, as they must be robust against link failures.

It is also possible to determine the robustness of a schedule by detecting or diagnosing latent hazards in computation-completed schedules. Atallah *et al.* [131] detected possible defects by injecting a single event upset (SEU) error into the generated schedule. This injection determines the weak point of the schedule for gate-blocking or message-lagging defects. Although this paper is specified as an SEU error scenario, it sufficiently represents defects caused by comparing the robustness of single path and multiple paths and manipulating schedule information. Likewise, machine learning-based fault detection frameworks to accelerate the schedule viability have also been devised because the scheduling fault diagnosis process usually occurs in pre-deployment stages [132], [133].

V. NETWORK MANAGEMENT

Those managing a large industrial automation network with hundreds of SWs must ensure accurate real-time requirements. Network devices can be clustered based on physical location or have separate hierarchies according to their role groups and can be divided into various domains. Different domains may imply different protocols and may also include various wireless components. To operate this network seamlessly, we need a flexible and scalable method for efficient network configuration and management, preferably automated.

This section deals with studies on network management in TSN, such as discovery, configuration, policy, control, clock synchronization, and GCL deployment specific to TSN. The potential of TSN combined with next-generation network technologies, such as network virtualization or data modeling for IT and operation technology (OT) convergence, is also presented. Finally, we introduce several case studies for the application of TSN in various industrial fields.

A. TIME SYNCHRONIZATION

Stanton [134] provided a tutorial on IEEE 802.1AS [19] time synchronization for TSN with a detailed background explanation including comparison of gPTP and PTP.

Beyond the standard, a few studies have proposed enhancements [135], [136] for more precise or reliable time synchronization with simulation-based performance comparison.

Zhu *et al*. [135] proposed a method for re-synchronization when the main clock is abnormal and a method for increasing synchronization accuracy by adjusting the synchronization interval and proposing a separate algorithm for sending synchronization messages. Furthermore, IEEE 802.1AS assumes symmetric communication links; therefore, Baniabdelghany *et al*. [136] proposed an extended synchronization protocol to compensate for asymmetric delays by measuring accurate delay values based on the neighbor rate ratio and cumulative scaled rate offset.

In addition to research for precision, another study proposes a software defined networking (SDN)-based solution that mitigates the complexity associated with the IEEE 802.1AS configuration and analyzes its performance using real-world devices [52]. For time synchronization in heterogeneous networks, the authors of [137] suggested a solution to integrate the clock synchronization methodology of EtherCAT and TSN with high precision and compatibility.

B. FRAMEWORK/MODELING

From AVB, IEEE 802.1Qat *Stream Reservation Protocol (SRP)* [138] features stream registration, reservation, and deregistration [139] procedures for the transmission path from talker to listener. This distributed method is less agile to changes in stream or network configuration than the centralized approach despite other advantages. Moreover, it is unsuitable for TSN because GCL scheduling requires information from the entire network and streams in advance. These are the motivations of the research work, such as a feasibility analysis framework [140], a modeling study for network configurations aiming for the ease of computing and deploying GCL schedules, and studies to support autoconfiguration (plug-and-play) mechanisms [141]–[144].

Accordingly, IEEE 802.1Qcc [21] defines *centralized network configuration (CNC)* and *centralized user configuration (CUC)* entities that support the NETCONF or RESTCONF protocols. The IEEE 802.1Qcp *YANG Data Model* [145] is proposed for enabling more flexible network configuration [146], [147]. As centralized control is regarded as a viable option in TSN, the use of SDN for TSN has also actively been studied [148]–[150]. Several solutions using SDN to implement fully centralized models of 802.1Qcc have been presented [41], [48], [49], [51], [151], [152].

In addition, heuristics that enable runtime management are presented, demonstrating its applicability in the fog computing platform [153], [154]. Moreover, IEC 62541 OPC-UA [155] is an industrial protocol and modeling standard for IT-OT convergence. Additionally, OPC-UA is expected to integrate with TSN by providing reliable network interoperability at OSI 5-7 layers, and its standardization is currently in progress (Fig. 2). In particular, OPC-UA Pub/Sub expands the existing server/client structure to a flexible structure for many-to-many connections, making it more useful in industrial machine-type networks. In [50], [156], [157], the authors introduced the integration of OPC-UA and TSN and discussed open issues and obstacles to be addressed for IT-OT convergence. Open62541[7] [158], node-opcua,[8] and python-opcua[9] are in development as open-source projects to implement OPC-UA stack, and research demonstrates TSN with OPC-UA using these projects [40], [55], [159].

C. CASE STUDY/PROOF-OF-CONCEPT

The in-vehicle network is one of the critical target domains of TSN. Motivated by typical mixed-critical automotive applications, Farzaneh and Knoll [160] tested the performance of TAS via actual experiments using three SWs with FPGA-based 802.1Qbv ports. Brunner *et al.* [161] foresaw the direction of future in-vehicle network architecture as the evolving automotive electrical/electronic architecture demands additional requirements, and highlighted the appropriateness of TSN's low latency, redundancy, and bandwidth restriction properties.

A larger portion of TSN work has been in many industrial domains that urge IT-OT integration convenience. Nsaibi *et al*. [33] composed a tree topology network that integrates TSN and Sercos and demonstrated the possibility and potential of integrating a legacy industrial Ethernet with TSN through simulations. Integration of TSN with next-generation technologies, such as IEEE 11073 service-oriented device connectivity [162] or openSAFETY [163] has also been researched [164], [165].

Sanchez-Garrido *et al.* [166] presented a use case of TSN, which implements a deterministic system in which all substations can communicate with each other through an Ethernet-based infrastructure. The authors installed the TSN system directly at a substation facility of Grupo Cuerva SL, one of Spain's electricity providers, and demonstrated that the TSN system satisfies IEC 61850 [167]

[7] https://github.com/open62541/open62541

[8] https://github.com/node-opcua/node-opcua

[9] https://github.com/FreeOpcUa/python-opcua

mixed-criticality traffic requirements well. Likewise, Docquier *et al.* [168] studied the use of the IEC 61850 protocol over TSN, and Muguira *et al.* [45] focused on the security of electrical sector digitalization through IEC 61850 over TSN.

Many case studies have been conducted to demonstrate the effectiveness of integrating OPC-UA and TSN. Li *et al.* [46] proposed a two-tier OPC-UA TSN communication architecture for a manufacturing system network and evaluated their prototype through experiments to demonstrate the feasibility of TSN for an industrial network.

Gogolev *et al.* demonstrated the basic concept of OPC-UA over TSN using COTS devices that are widely used in the industrial field, such as a laser level or gauge pressure transmitter. The authors revealed that when OPC-UA is used with TSN, communication becomes much faster and stabler even under high network congestion [43]. They also indicated that OPC-UA communication latency could be reduced even if there is no TSN support in end devices, instead only using traffic shaping (TAS and CBS) of TSN-enabled SWs, (i.e., without additional updates for end-devices) [44].

Golatowski *et al.* [169] used TSN for deterministic communication of system control tasks (crane-crane remote controller) using the OPC-UA framework and evaluated their proof-of-concept. In contrast, Agarwal *et al.* [170] investigated integrating a data distribution service [171], a distributed system information management technology similar to OPC-UA, with TSN.

D. WIRELESS INTEGRATION

Cavalcanti *et al.* described the requirements (e.g., time synchronization, reliability, scheduling, and network management) in detail when applying TSN to wireless systems. They analyzed the challenges that arose while performing TSN features through wireless networks (5G, Wi-Fi, ZigBee, etc.) [172].

Seijo *et al.* analyzed the potential problems in the PTP synchronization mechanism due to wireless characteristics, such as sampling and multipath propagation. They proposed an improved wireless timestamping method robust to these problems [173].

Shrestha *et al.* [174] also conducted a study to improve the performance of PTP in wireless by estimating the clock drift factor considering wireless characteristics. They included it in the clock offset estimation.

Romanov *et al.* [175] focused on Wi-Fi to enable time synchronization in a wired/wireless hybrid network where the TSN backbone and Wi-Fi cells are mixed by modifying the synchronization algorithm without re-implementing the MAC layer. A new QoS access category was introduced to improve the delay and jitter performance of Wi-Fi when integrating the Wi-Fi into a TSN network [176].

The 3rd Generation Partnership Project (3GPP) recently finalized 5G release 16, providing ultra-reliability and low-latency communication services for the industrial Internet of things (IIoT). This release was a step closer to core wireless technology for factory automation, such as motion control or control-to-control.

Neumann *et al.* presented various scenarios for combining 5G and the industrial Ethernet [177]. Further, Khoshnevisan *et al.* covered the overview of TSN and 5G integration for industrial factory automation, presenting a prototype system to validate the potential for using an integrated 5G New Radio system with PROFINET (although not TSN, but considering the possibility of substitution in the future) [178].

Furthermore, some researchers [179]–[182] have analyzed the integration of 5G systems as a logical TSN bridge in a TSN network environment. Particularly, a system-level simulator was presented in [179] to analyze the effects of a 5G system supporting ST in TSN, emphasizing the necessity for more research for seamless integration. Moreover, 5G and TSN-integrated time synchronization mechanisms have been analyzed [42], [182].

In addition to the perspective of 5G as an industrial Ethernet link, the incorporation of TSN to enhance the 5G mobile network functionality has also been studied. A change in the structure of the existing fronthaul is required to handle the tremendous increase in traffic expected when switching to 5G. Therefore, conversion from the circuit-based common public radio interface to the packet-switching-based common public radio interface has also been studied, and TSNs are expected to play an essential role in the next fronthaul architecture.

The TSN Task Group standardized the IEEE 802.1CM *Time-Sensitive Networking for Fronthaul* [183]. It provides profiles for the time-sensitive fronthaul and specifies synchronization, scheduling, and preemption mechanisms.

Bhattacharjee *et al.* introduced the 802.1CM overview. They evaluated the performance of IEEE 802.1Qbv and Qbu with real fronthaul traffic [184].

Liu *et al.* proposed a simple and effective heuristic 'higher rate flow scheduled later' scheme for flow scheduling in fronthaul networks with detailed background knowledge of 5G fronthaul [185]. In addition, TAS enhancement [25] and asynchronous scheduling approaches [186] have been studied for deterministic transmission in 5G fronthaul networks.

Furthermore, Ansah *et al.* adopted the concept of network slicing of 5G. This concept was to implement the logical separation of various services with different requirements and demonstrate a simulation-based analysis for the worst-case end-to-end latency of each service [187].

VI. FOR TSN RESEARCHERS

In this category, we cover a few subtopics that readers who wish to research TSN should know before addressing the cutting-edge of this field. We introduce simulators regarding which frameworks are widely used or which modules are available, depending on the specific features requiring simulation. Furthermore, as TSN demands more 'real' assessments of its proposed extensions and improvements, studies demonstrating implementation and evaluation of

TSN capabilities on real hardware provide references concerning the currently available devices and limitations.

A. SIMULATOR

In 2016, Heise *et al.* proposed TSimNet [188], a TSN simulator that extends INET to modularly design non-time-based TSN features, such as FP, per-stream policing, and FRER (802.1Qbu, Qci, and CB). This study was before the official standardization of TAS in 2018. Soon after, in 2018, Jiang *et al.* [47], [189] implemented time-based features, such as IEEE 802.1AS for time synchronization and the IEEE 802.1Qbv TAS simulation model, by extending CoRE4INET [190].

In 2019, Falk *et al.* developed NeSTiNg [26], a TSN framework based on INET, including IEEE 802.1Qbv, Qbu, Qav, and VLAN tagging. Next, Martenvormfelde *et al.* modeled 5G as a logical TSN bridge in addition to the NeSTiNg framework in 2020 [191].

Riverbed [23] (formerly known as OPNET) has also been widely used for developing TSN models since 2017. According to our survey, research on modules by function has continued: IEEE 802.1AS [192], 1Qbv [193], [194], 1Qci [194], and 1CB [195]. Riverbed is also a reliable simulator used to design complex communication protocols and large-scale networks. Another study analyzed TSN scheduling using UPPAAL [196], a well-defined timed model checker [197].

B. HARDWARE DESIGN

Software-level timestamping has difficulty achieving the precision required by time-triggered Ethernet protocols; thus, support from designated hardware is required. Kyriakakis *et al.* [198] proposed an architecture that includes a PTP hardware-assist unit in an FPGA-based multicore platform to improve the precision of timestamps. They analyzed the worst-case execution time through experiments.

Groß *et al.* [199] proposed a flexible and scalable architecture that can run entirely in the hardware, entirely in software, or both. This selection is so that certain features can select an appropriate method according to their timing requirements.

Coleman *et al.* [200] demonstrated that the performance of TSN Ethernet could degrade depending on the synchronization accuracy between the network clock and CPU clock. For a more sophisticated synchronization, the authors proposed an improvement using PCIe. They evaluated the improvement by experimenting with COTS hardware.

Pruski *et al.* [201] revealed that the hardware for the existing switch queuing system has a fundamental problem, limiting the performance of TAS, CBS, and FP. They proposed a new SW architecture to compensate for this.

Li *et al.* [202] proposed an efficient and fault-tolerant memory architecture to facilitate complex TSN functions, even within limited SW memory. Lastly, Vlk *et al.* [35] proposed a method to detect and handle frames sent differently from the schedule in the SW, relaxing the schedule constraints and increasing schedulability.

C. MISCELLANEOUS

Many concepts are confused among TSN and TTEthernet; thus, it is vital to understand their differences. Zhao *et al.* [10] presented a comparative analysis of these in various aspects, such as network architecture, clock synchronization mechanism, and redundancy approach. Steiner *et al.* [31] compared the transmission mechanism of traditional time-triggered switching and IEEE 802.1Qbv TAS. We encourage the reader to refer to the mentioned tutorials to understand TSN overall [4]–[6], [24].

Network calculus is frequently used to compute the upper bounds on the latency and buffer size of various queuing and scheduling mechanisms in TSN. Maile *et al.* [203] surveyed studies that present network calculus models for TSN and analyzed their differences.

The future direction of TSN applications in each industry is also of great importance. Samii and Zinner [204] reviewed the suitability of TSN as an internal network for autonomous driving systems taking the five levels defined by SAE J3016 as a use case [205]. The authors introduced switched Ethernet technologies enhanced by several critical attributes of TSN and mapped the requirements for future automotive systems. Hallmans *et al.* [206] discussed the main challenges and issues of using TSN technology in long-life industrial distributed control systems, such as power plants and smart energy transmission systems, and proposed strategies for designing of these systems.

VII. FUTURE CHALLENGES

This article describes literature reporting features making TSN different from the traditional Ethernet, namely the *TSN core functions*. We cover the studies on *network management* that enable these powerful features to be exquisite, flexible, and interoperable. Subsequently, we survey several reference studies *for TSN Researchers*, hoping that more diverse and novel TSN research will follow. In this section, based on the efforts made so far, we outline future challenges that TSN technology must address to become more mature for practical large-scale deployment and use.

A. FLEXIBLE TRAFFIC CLASS

The TSN achieves low-latency and fault-tolerance through traffic prioritizing, streamifying, shaping, and policing. Traffic in TSN is classified according to the requirements and configuration known a priori, and bounded latency is guaranteed through suitable shapers assigned to each class or stream in SWs. All is well in a fully prepared state, but TSN is vulnerable to unexpected runtime changes in network conditions. To compensate for this vulnerability, TSN requires a method to recognize network status changes in real-time and seamlessly modify the traffic class configuration on-line without halting or pausing the system.

For example, due to an out-of-sync clock or unexpected change in the network topology, the schedule may become meaningless, or a new schedule may need to be synthesized

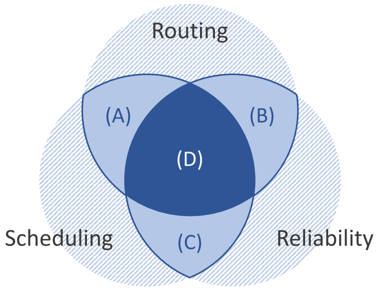


FIGURE 8. Classification of routing, scheduling, and reliability related studies.

(which takes a long time). If the class of ST flows can adaptively switch to (for example) Class A so that ATS or CBS regulates ST flows without regenerating and redeploying a new schedule, it may still be possible to guarantee a certain degree of QoS. Additionally, in preparation for this situation, the network designer can reserve a separate class subject to ST's urgent downgrade with a lower priority lower than ST but higher than Class A. This method has no side effects on lower classes because there is no traffic of that class under normal circumstances, and the TSN group allows the AVB classes to be completely customized. Furthermore, data analytics and ML-assisted algorithms are becoming more important to realize online solutions that enable the TSN network to adapt to the changing traffic demands and network conditions.

B. ROUTING FOR TIME-AWARE SHAPER SCHEDULING AND RELIABILITY

Routing is an indispensable and vital process for network systems, and TSN is no exception. Shortest-path bridging is the intuitive default for low end-to-end latency of critical traffic. Alternatively, equal-cost multipath is an exemplary routing strategy for load balance. Among TSN standards, IEEE 802.1Qca *Path Control and Reservation* [207] is an IS-IS protocol that provides a set of routing functions to manage bridged networks.

There are many potential disadvantages when multiple critical flows select the same path. Therefore, by considering the routing problem jointly in SSA for time-triggered transmission of ST traffic, it may be possible to obtain a more optimized schedule (e.g., minimizing flowspan and maximizing ST window utilization). Not just for SSA, the importance of routing in TSN research can also be emphasized from a reliability perspective, such as multipath routing for redundancy to address link or device failures.

Fig. 8 is a diagram for understanding the coupling of routing, scheduling, and reliability at a glance. We identify the shared focus areas for the main interests of relevant studies in each part and then discuss the most important challenges.

For example, studies that handle routing and scheduling simultaneously belong to Part A, and Part B includes papers addressing both reliability and routing issues by, for example, using spatial redundancy. Studies in Part C are mostly GCL schedule synthesis techniques considering ST with fault tolerance for reliability. Although studies in Part C do not consider routing, some use temporal redundancy in the scheduling stage or verify the schedule stability. Studies in Part D concurrently consider all three characteristics of routing, scheduling and reliability. In other words, studies in Part A or Part B focus on achieving each goal of scheduling and reliability, respectively, by combining appropriate routing information, whereas Part D includes studies dealing with both simultaneously. Two studies by Atallah *et al*. [30], [127] are good examples of Part D, which address both scheduling and reliability by considering proper routing, and we believe that there should be more studies going forward.

C. INTEGRATION WITH OTHER PROTOCOLS OR TECHNOLOGIES

The accuracy of scheduled transmission depends on how sharp and agile each window in the schedule timeline responds to its offset. However, research on stability and recovery, such as the fault handling process of the clock system, is still insufficient. In addition, IEEE 802.1AS specifies a protocol that provides accurate synchronization time on the links for the Ethernet passive optical network (EPON) and coordinated shared network (CSN). However, no research has been conducted on these. All studies have only been subjected to Ethernet full-duplex point-to-point links (minimally for 802.11 wireless).

The problem of TSN research focusing on Ethernet is the same in *time synchronization* and areas that require interaction with external networks, such as *framework/modeling* or *wireless integration*. Some studies evaluating the integration of TSN with a powerful communication middleware called OPC-UA [155] to provide interoperability between systems have been preferentially conducted, but this is a very basic step. We emphasize the importance of evaluating interfaces for compatibility between heterogeneous network protocols, including OPC-UA. Appropriate north/southbound, east/westbound interfaces should be implemented, and a benchmark should be evaluated.

The need for such interoperability evaluation studies must aim particularly at even more complex wireless communication systems. Smart factories in the future will be wirelessly connect various last-hop devices, such as actuators and robots, for mobility and scalability through 5G NR, Wi-Fi, wirelessHART, ZigBee, Bluetooth, and others. Therefore, related research is ongoing in each field, as it must be rapidly and precisely controlled for industrial purposes. This interoperability evaluation is also necessary for external vehicle networks, such as dedicated short range communication (DSRC), wireless access in vehicular environment (WAVE), and cellular vehicle-to-everything (C-V2X).

D. NETWORK VIRTUALIZATION

These futuristic wireless networks can secure deterministic communication, flexibility, scalability, and convenience of configuration by decoupling through hardware virtualization. They aim to provide customized QoS by multiplexing virtualized independent logical networks in a single physical network infrastructure. The 5G network slicing and SDN network function virtualization are prevailing types of these virtualizations. Access networks can provide virtualized resources in their own way, and core networks slice, share, or isolate them and provide control as needed in software. From this viewpoint, the virtual LAN concept of TSN serves as the nucleus of the core network. Thus, more realistic research on contact points with various access networks is needed.

E. FULLY FEATURED SIMULATOR AND HARDWARE

We learned the key differences between NeSTiNg and CoRE4INET, the two most recently and widely used TSN simulators, by using them in our research. We realized that more TSN features must be incorporated, and some practical implementation-specific restrictions limit what can be simulated. For example, NeSTiNg is only available in a Linux environment, but CoRE4INET is available in Windows and Linux. Both NeSTiNg and CoRE4INET use a configuration file to set the routes and GCL but in different formats. The NeSTiNg models queue loss by importing INET code, but CoRE4INET has its own queue loss code which requires customizing. Moreover, NeSTiNg follows the priority-to-traffic class mapping table in the IEEE 802.1Q standard (Class A: 2 and Class B: 3). However, CoRE4INET maps priority and traffic class numbers one-to-one (Class A: 5 and Class B: 6) Thus, TSN simulators with more TSN features and realistic data and environment models are required to evaluate TSN and its research more accurately.

Finally, fully featured COTS equipment (e.g. SW, CNC) and development/evaluation kits for academic research are lacking in TSN. Similar to the simulators, more TSN features must be incorporated, and some implementation-specific restrictions limit their capabilities. We have emphasized several times throughout the paper the need for better equipment and development environment for researchers to study TSN. Only then would we be able to take TSN to the next level for real industrial deployments.

VIII. CONCLUSION

Tremendous efforts on open standards for the integrated real-time networking protocol, TSN, are underway in industry and academia. In addition to the functional strengths of ultra-low latency/jitter and reliability, the fact that it is based on the most widely used Ethernet, makes TSN economical in many ways let alone interoperability. With these advantages, TSN is expected to become the de facto next-generation LAN infrastructure that will be the foundation of near-future leading technologies, such as Industry 4.0, smart factories, intelligent transportation system, and 5G. Throughout this paper, we conducted a comprehensive statistical survey of 174 TSN publications, the first study of its kind to the best of our knowledge. We have examined in a timely manner what topics have been researched so far in TSN, their trends, and what is lacking and needed. We aim for this paper to serve as a pointer and directional guide to many fellow researchers who wish to study TSN in the future.

APPENDIX–LIST OF ACRONYMS

3GPP	3rd Generation Partnership Project.
AFDX	Avionics Full-Duplex Switched Ethernet.
ASIC	Application-Specific Integrated Circuit.
ATS	Asynchronous Traffic Shaper.
AVB	Audio/Video Bridging.
CAN	Controller Area Network.
CBS	Credit-Based Shaper.
CDF	Cumulative Distribution Function.
CEV	Crew Exploration Vehicle.
CNC	Centralized Network Configuration.
COTS	Commercial off-the-shelf.
CQF	Cyclic Queuing and Forwarding.
CSN	Coordinated Shared Network.
CUC	Centralized User Configuration.
C-V2X	Cellular Vehicle-to-Everything.
DetNET	Deterministic Network.
DSRC	Dedicated Short Range Communication.
EPON	Ethernet Passive Optical Network.
ES	End System.
EtherCAT	Ethernet for Control Automation Technology.
FTT-Ethernet	Flexible Time-Triggered Ethernet.
FP	Frame Preemption.
FPGA	Field Programmable Gate Array.
FRER	Frame Replication and Elimination for Reliability.
GB	Guard Band.
GCL	Gate Control List.
gPTP	Generic Precision Time Protocol.
HSR	High Availability and Seamless Redundancy.
IIoT	Industrial Internet of Things.
ILP	Integer Linear Programming.
IoT	Internet of Things.
IT	Information Technology.
OPC-UA	Open Platform Communications Unified Architecture.
OT	Operation Technology.
PRP	Parallel Redundancy Protocol.
PS	Paternoster Scheduling.
PTP	Precision Time Protocol.
QoS	Quality-of-Service.
SAE	Society of Automotive Engineers.
SDN	Software Defined Networking.
SEU	Single Event Upset.
SMT	Satisfiability Modulo Theories.
SoC	Systems-on-a-Chip.
SRP	Stream Reservation Protocol.
SSA	Scheduling Synthesis Approach.
ST	Scheduled Traffic.
SW	Switch.

TAS	Time-Aware Shaper.
TSN	Time-Sensitive Networking.
TTEthernet	Time-Triggered Ethernet.
UBS	Urgency-Based Scheduler.
WAVE	Wireless Access in Vehicular Environment.

REFERENCES

[1] *IEEE Standard for Ethernet*, Standard 802.3-2018, 2018, pp. 1–5600.

[2] (2017). *IEEE Time-Sensitive Networking Task Group*. Accessed: Jul. 2021. [Online]. Available: http://www.ieee802.org/1/pages/tsn.html

[3] *IEEE Standard for Local and Metropolitan Area Network–Bridges and Bridged Networks*, Standard 802.1Q-2018, 2018, pp. 1–1993.

[4] A. Nasrallah, A. S. Thyagaturu, Z. Alharbi, C. Wang, X. Shao, M. Reisslein, and H. ElBakoury, "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., 2019.

[5] J. L. Messenger, "Time-sensitive networking: An introduction," *IEEE Commun. Standards Mag.*, vol. 2, no. 2, pp. 29–33, Jun. 2018.

[6] N. Finn, "Introduction to time-sensitive networking," *IEEE Commun. Standards Mag.*, vol. 2, no. 2, pp. 22–28, Jun. 2018.

[7] *IEEE Standard for Local and Metropolitan Area Networks–Audio Video Bridging (AVB) Systems*, Standard 802.1BA-2011, 2011, pp. 1–45.

[8] W. Steiner, G. Bauer, B. Hall, M. Paulitsch, and S. Varadarajan, "TTEthernet dataflow concept," in *Proc. 8th IEEE Int. Symp. Netw. Comput. Appl.*, Jul. 2009, pp. 319–322.

[9] M. D. J. Teener, A. N. Fredette, C. Boiger, P. Klein, C. Gunther, D. Olsen, and K. Stanton, "Heterogeneous networks for audio and video: Using IEEE 802.1 audio video bridging," *Proc. IEEE*, vol. 101, no. 11, pp. 2339–2354, Nov. 2013.

[10] L. Zhao, F. He, E. Li, and J. Lu, "Comparison of time sensitive networking (TSN) and TTEthernet," in *Proc. IEEE/AIAA 37th Digit. Avionics Syst. Conf. (DASC)*, Sep. 2018, pp. 1–7.

[11] *IEEE Standard for Local and Metropolitan Area Networks–Bridges and Bridged Networks–Amendment 25: Enhancements for Scheduled Traffic*, Standard 802.1Qbv-2015, 2016, pp. 1–57.

[12] *IEEE Standard for Local and Metropolitan Area Networks–Virtual Bridged Local Area Networks Amendment 12: Forwarding and Queuing Enhancements for Time-Sensitive Streams*, Standard 802.1Qav-2009, 2010, pp. 1–72.

[13] *IEEE Standard for Local and Metropolitan Area Networks–Bridges and Bridged Networks–Amendment 34: Asynchronous Traffic Shaping*, Standard 802.1Qcr-2020, 2020, pp. 1–151.

[14] *IEEE Standard for Local and Metropolitan Area Networks–Bridges and Bridged Networks–Amendment 26: Frame Preemption*, Standard 802.1Qbu-2016, 2016, pp. 1–52.

[15] *IEEE Standard for Ethernet Amendment 5: Specification and Management Parameters for Interspersing Express Traffic*, Standard 802.3br-2016, 2016, pp. 1–58.

[16] H. Kirrmann, K. Weber, O. Kleineberg, and H. Weibel, "HSR: Zero recovery time and low-cost redundancy for industrial Ethernet (high availability seamless redundancy, IEC 62439-3)," in *Proc. IEEE Conf. Emerg. Technol. Factory Autom.*, Sep. 2009, pp. 1–4.

[17] H. Kirrmann, M. Hansson, and P. Muri, "IEC 62439 PRP: Bumpless recovery for highly available, hard real-time industrial networks," in *Proc. IEEE Conf. Emerg. Technol. Factory Autom. (EFTA)*, Sep. 2007, pp. 1396–1399.

[18] *IEEE Standard for Local and Metropolitan Area Networks–Frame Replication and Elimination for Reliability*, Standard 802.1CB-2017, 2017, pp. 1–102.

[19] *IEEE Standard for Local and Metropolitan Area Networks–Timing and Synchronization for Time-Sensitive Applications*, Standard 802.1AS-2020, 2020, pp. 1–421.

[20] *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*, Standard 1588-2008, 2008, pp. 1–300.

[21] *IEEE Standard for Local and Metropolitan Area Networks–Bridges and Bridged Networks–Amendment 31: Stream Reservation Protocol (SRP) Enhancements and Performance Improvements*, Standard 802.1Qcc-2018, 2018, pp. 1–208.

[22] *OMNeT++ Discrete Event Simulator*. Accessed: Jul. 2021. [Online]. Available: https://omnetpp.org/

[23] *Riverbed's Modeler Academic Community*. Accessed: Jul. 2021. [Online]. Available: https://cms-api.riverbed.com/portal/community_home

[24] L. Lo Bello and W. Steiner, "A perspective on IEEE time-sensitive networking for industrial communication and automation systems," *Proc. IEEE*, vol. 107, no. 6, pp. 1094–1120, Jun. 2019.

[25] D. Hisano, Y. Nakayama, T. Kubo, T. Shimizu, H. Nakamura, J. Terada, and A. Otaka, "Gate-shrunk time aware shaper: Low-latency converged network for 5G fronthaul and M2M services," in *Proc. GLOBECOM IEEE Global Commun. Conf.*, Dec. 2017, pp. 1–6.

[26] J. Falk, D. Hellmanns, B. Carabelli, N. Nayak, F. Durr, S. Kehrer, and K. Rothermel, "NeSTiNg: Simulating IEEE time-sensitive networking (TSN) in OMNeT++," in *Proc. Int. Conf. Netw. Syst. (NetSys)*, Mar. 2019, pp. 1–8.

[27] R. Serna Oliver, S. S. Craciunas, and W. Steiner, "IEEE 802.1Qbv gate control list synthesis using array theory encoding," in *Proc. IEEE Real-Time Embedded Technol. Appl. Symp. (RTAS)*, Apr. 2018, pp. 13–24.

[28] J. Falk, F. Durr, and K. Rothermel, "Exploring practical limitations of joint routing and scheduling for TSN with ILP," in *Proc. IEEE 24th Int. Conf. Embedded Real-Time Comput. Syst. Appl. (RTCSA)*, Aug. 2018, pp. 136–146.

[29] A. Alnajim, S. Salehi, and C.-C. Shen, "Incremental path-selection and scheduling for time-sensitive networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.

[30] A. A. Atallah, G. B. Hamad, and O. A. Mohamed, "Routing and scheduling of time-triggered traffic in time-sensitive networks," *IEEE Trans. Ind. Informat.*, vol. 16, no. 7, pp. 4525–4534, Jul. 2020.

[31] W. Steiner, S. S. Craciunas, and R. S. Oliver, "Traffic planning for time-sensitive communication," *IEEE Commun. Standards Mag.*, vol. 2, no. 2, pp. 42–47, Jun. 2018.

[32] M. Gutierrez, W. Steiner, R. Dobrin, and S. Punnekkat, "Synchronization quality of IEEE 802.1AS in large-scale industrial automation networks," in *Proc. IEEE Real-Time Embedded Technol. Appl. Symp. (RTAS)*, Apr. 2017, pp. 273–282.

[33] S. Nsaibi, L. Leurs, and H. D. Schotten, "Formal and simulation-based timing analysis of industrial-Ethernet sercos III over TSN," in *Proc. IEEE/ACM 21st Int. Symp. Distrib. Simulation Real Time Appl. (DS-RT)*, Oct. 2017, pp. 1–8.

[34] D. Hellmanns, J. Falk, A. Glavackij, R. Hummen, S. Kehrer, and F. Durr, "On the performance of stream-based, class-based time-aware shaping and frame preemption in TSN," in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, Feb. 2020, pp. 298–303.

[35] M. Vlk, Z. Hanzalek, K. Brejchova, S. Tang, S. Bhattacharjee, and S. Fu, "Enhancing schedulability and throughput of time-triggered traffic in IEEE 802.1Qbv time-sensitive networks," *IEEE Trans. Commun.*, vol. 68, no. 11, pp. 7023–7038, Nov. 2020.

[36] A. Gogolev and P. Bauer, "A simpler TSN? Traffic scheduling vs. preemption," in *Proc. 25th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2020, pp. 183–189.

[37] M. Paulitsch, E. Schmidt, C. Scherrer, and H. Kantz, "Time-triggered communication," in *Industrial Applications*, 1st ed., R. Obermaisser, Ed. Boca Raton, FL, USA: CRC Press, 2012, ch. 14, doi: 10.1201/9781315217512.

[38] L. Zhao, P. Pop, Z. Gong, and B. Fang, "Improving latency analysis for flexible window-based GCL scheduling in TSN networks by integration of consecutive nodes offsets," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5574–5584, Apr. 2021.

[39] *INET Framework*. Accessed: Jul. 2021. [Online]. Available: https://inet.omnetpp.org/

[40] J. Pfrommer, A. Ebner, S. Ravikumar, and B. Karunakaran, "Open source OPC UA PubSub over TSN for realtime industrial communication," in *Proc. IEEE 23rd Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2018, pp. 1087–1090.

[41] S. B. H. Said, Q. H. Truong, and M. Boc, "SDN-based configuration solution for IEEE 802.1 time sensitive networking (TSN)," *ACM SIGBED Rev.*, vol. 16, no. 1, pp. 27–32, 2019.

[42] M. Gundall, C. Huber, P. Rost, R. Halfmann, and H. D. Schotten, "Integration of 5G with TSN as prerequisite for a highly flexible future industrial automation: Time synchronization based on IEEE 802.1AS," in *Proc. IECON 46th Annu. Conf. IEEE Ind. Electron. Soc.*, Oct. 2020, pp. 3823–3830.

[43] A. Gogolev, F. Mendoza, and R. Braun, "TSN-enabled OPC UA in field devices," in *Proc. IEEE 23rd Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2018, pp. 297–303.

[44] A. Gogolev, R. Braun, and P. Bauer, ''TSN traffic shaping for OPC UA field devices,'' in *Proc. IEEE 17th Int. Conf. Ind. Informat. (INDIN)*, Jul. 2019, pp. 951–956.

[45] L. Muguira, J. Lazaro, S. Alonso, A. Astarloa, and M. Rodriguez, ''Secure critical traffic of the electric sector over time-sensitive networking,'' in *Proc. 35th Conf. Design Circuits Integr. Syst. (DCIS)*, Nov. 2020, pp. 1–6.

[46] Y. Li, J. Jiang, C. Lee, and S. H. Hong, ''Practical implementation of an OPC UA TSN communication architecture for a manufacturing system,'' *IEEE Access*, vol. 8, pp. 200100–200111, 2020.

[47] J. Jiang, Y. Li, S. H. Hong, M. Yu, A. Xu, and M. Wei, ''A simulation model for time-sensitive networking (TSN) with experimental validation,'' in *Proc. 24th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2019, pp. 153–160.

[48] M. Bohm, J. Ohms, and D. Wermser, ''Multi-domain time-sensitive networks—An east-westbound protocol for dynamic TSN-stream configuration across domains,'' in *Proc. 24th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2019, pp. 1363–1366.

[49] M. Boehm, J. Ohms, M. Kumar, O. Gebauer, and D. Wermser, ''Time-sensitive software-defined networking: A unified control-plane for TSN and SDN,'' in *Proc. Mobile Commun.-Technol. Appl. ITG-Symp.*, May 2019, pp. 1–6.

[50] T. Kobzan, I. Blocher, M. Hendel, S. Althoff, A. Gerhard, S. Schriegel, and J. Jasperneite, ''Configuration solution for TSN-based industrial networks utilizing SDN and OPC UA,'' in *Proc. 25th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2020, pp. 1629–1636.

[51] T. Gerhard, T. Kobzan, I. Blocher, and M. Hendel, ''Software-defined flow reservation: Configuring IEEE 802.1Q time-sensitive networks by the use of software-defined networking,'' in *Proc. 24th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2019, pp. 216–223.

[52] M.-T. Thi, S. Ben Hadj Said, and M. Boc, ''SDN-based management solution for time synchronization in TSN networks,'' in *Proc. 25th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2020, pp. 361–368.

[53] C. Basumatary and H. S. Satheesh, ''Queuing policies for enhanced latency in time sensitive networking,'' in *Proc. IEEE Int. Conf. Electron., Comput. Commun. Technol. (CONECCT)*, Jul. 2020, pp. 1–7.

[54] I. Alvarez, L. Moutinho, P. Pedreiras, D. Bujosa, J. Proenza, and L. Almeida, ''Comparing admission control architectures for real-time Ethernet,'' *IEEE Access*, vol. 8, pp. 105521–105534, 2020.

[55] C. Eymuller, J. Hanke, A. Hoffmann, M. Kugelmann, and W. Reif, ''Real-time capable OPC-UA programs over TSN for distributed industrial control,'' in *Proc. 25th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2020, pp. 278–285.

[56] S. S. Craciunas, R. S. Oliver, M. Chmelík, and W. Steiner, ''Scheduling real-time communication in IEEE 802.1Qbv time sensitive networks,'' in *Proc. 24th Int. Conf. Real-Time Netw. Syst. (RTNS)*, 2016, pp. 183–192.

[57] S. S. Craciunas and R. S. Oliver, ''Combined task- and network-level scheduling for distributed time-triggered systems,'' *Real-Time Syst.*, vol. 52, no. 2, pp. 161–200, Mar. 2016.

[58] A. C. T. D. Santos, B. Schneider, and V. Nigam, ''TSNSCHED: Automated schedule generation for time sensitive networking,'' in *Proc. Formal Methods Comput. Aided Design (FMCAD)*, Oct. 2019, pp. 69–77.

[59] X. Jin, C. Xia, N. Guan, C. Xu, D. Li, Y. Yin, and P. Zeng, ''Real-time scheduling of massive data in time sensitive networks with a limited number of schedule entries,'' *IEEE Access*, vol. 8, pp. 6751–6767, 2020.

[60] X. Dai, S. Zhao, Y. Jiang, X. Jiao, X. S. Hu, and W. Chang, ''Fixed-priority scheduling and controller co-design for time-sensitive networks,'' in *Proc. 39th Int. Conf. Comput.-Aided Design*, Nov. 2020, pp. 1–9.

[61] N. Reusch, L. Zhao, S. S. Craciunas, and P. Pop, ''Window-based schedule synthesis for industrial IEEE 802.1Qbv TSN networks,'' in *Proc. 16th IEEE Int. Conf. Factory Commun. Syst. (WFCS)*, Apr. 2020, pp. 1–4.

[62] C.-C. Chuang, T.-H. Yu, C.-W. Lin, A.-C. Pang, and T.-J. Hsieh, ''Online stream-aware routing for TSN-based industrial control systems,'' in *Proc. 25th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2020, pp. 254–261.

[63] E. Schweissguth, P. Danielis, D. Timmermann, H. Parzyjegla, and G. Mühl, ''ILP-based joint routing and scheduling for time-triggered networks,'' in *Proc. 25th Int. Conf. Real-Time Netw. Syst.*, Oct. 2017, pp. 8–17.

[64] N. G. Nayak, F. Dürr, and K. Rothermel, ''Incremental flow scheduling and routing in time-sensitive software-defined networks,'' *IEEE Trans. Ind. Informat.*, vol. 14, no. 5, pp. 2066–2075, May 2017.

[65] R. Mahfouzi, A. Aminifar, S. Samii, A. Rezine, P. Eles, and Z. Peng, ''Stability-aware integrated routing and scheduling for control applications in Ethernet networks,'' in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2018, pp. 682–687.

[66] N. G. Nayak, F. Dürr, and K. Rothermel, ''Routing algorithms for IEEE802.1Qbv networks,'' *ACM SIGBED Rev.*, vol. 15, no. 3, pp. 13–18, Jun. 2018.

[67] V. Gavriluţ, L. Zhao, M. L. Raagaard, and P. Pop, ''AVB-aware routing and scheduling of time-triggered traffic for TSN,'' *IEEE Access*, vol. 6, pp. 75229–75243, 2018.

[68] R. Mahfouzi, A. Aminifar, S. Samii, P. Eles, and Z. Peng, ''Security-aware routing and scheduling for control applications on Ethernet TSN networks,'' *ACM Trans. Design Autom. Electron. Syst.*, vol. 25, no. 1, pp. 1–26, Jan. 2020.

[69] M. Pahlevan, N. Tabassam, and R. Obermaisser, ''Heuristic list scheduler for time triggered traffic in time sensitive networks,'' *SIGBED Rev.*, vol. 16, no. 1, pp. 15–20, Feb. 2019.

[70] Q. Yu and M. Gu, ''Adaptive group routing and scheduling in multicast time-sensitive networks,'' *IEEE Access*, vol. 8, pp. 37855–37865, 2020.

[71] A. Arestova, K. S. J. Hielscher, and R. German, ''Design of a hybrid genetic algorithm for time-sensitive networking,'' in *Proc. Int. Conf. Meas., Modeling Eval. Comput. Syst.* Cham, Switzerland: Springer, 2020, pp. 99–117.

[72] E. Schweissguth, D. Timmermann, H. Parzyjegla, P. Danielis, and G. Muhl, ''ILP-based routing and scheduling of multicast realtime traffic in time-sensitive networks,'' in *Proc. IEEE 26th Int. Conf. Embedded Real-Time Comput. Syst. Appl. (RTCSA)*, Aug. 2020, pp. 1–11.

[73] Q. Yu, H. Wan, X. Zhao, Y. Gao, and M. Gu, ''Online scheduling for dynamic VM migration in multicast time-sensitive networks,'' *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 3778–3788, Jun. 2020.

[74] G. N. Kumar, K. Katsalis, and P. Papadimitriou, ''Coupling source routing with time-sensitive networking,'' in *Proc. IEEE IFIP Netw. Conf. Netw.*, Jun. 2020, pp. 797–802.

[75] P. Pop, M. L. Raagaard, S. S. Craciunas, and W. Steiner, ''Design optimisation of cyber-physical distributed systems using IEEE time-sensitive networks,'' *IET Cyber-Phys. Syst., Theory Appl.*, vol. 1, no. 1, pp. 86–94, 2016.

[76] S. M. Laursen, P. Pop, and W. Steiner, ''Routing optimization of AVB streams in TSN networks,'' *SIGBED Rev.*, vol. 13, no. 4, pp. 43–48, Nov. 2016.

[77] A. A. Atallah, G. B. Hamad, and O. A. Mohamed, ''Multipath routing of mixed-critical traffic in time sensitive networks,'' in *Proc. Int. Conf. Ind., Eng. Other Appl. Appl. Intell. Syst.* Cham, Switzerland: Springer, 2019, pp. 504–515.

[78] F. Smirnov, M. Glaß, F. Reimann, and J. Teich, ''Optimizing message routing and scheduling in automotive mixed-criticality time-triggered networks,'' in *Proc. 54th Annu. Design Autom. Conf.*, Jun. 2017, pp. 1–6.

[79] V. Gavrilut and P. Pop, ''Scheduling in time sensitive networks (TSN) for mixed-criticality industrial applications,'' in *Proc. 14th IEEE Int. Workshop Factory Commun. Syst. (WFCS)*, Jun. 2018, pp. 1–4.

[80] V. Gavriluţ and P. Pop, ''Traffic-type assignment for TSN-based mixed-criticality cyber-physical systems,'' *ACM Trans. Cyber-Phys. Syst.*, vol. 4, no. 2, pp. 1–27, Feb. 2020.

[81] F. Dürr and N. G. Nayak, ''No-wait packet scheduling for IEEE time-sensitive networks (TSN),'' in *Proc. 24th Int. Conf. Real-Time Netw. Syst. (RTNS)*, 2016, pp. 203–212.

[82] F. Heilmann and G. Fohler, ''Size-based queuing: An approach to improve bandwidth utilization in TSN networks,'' *ACM SIGBED Rev.*, vol. 16, no. 1, pp. 9–14, Feb. 2019.

[83] *IEEE Standard for Local and Metropolitan Area Networks–Bridges and Bridged Networks–Amendment 29: Cyclic Queuing and Forwarding*, Standard 802.1Qch-2017, 2017, pp. 1–30.

[84] *IEEE Standard for Local and Metropolitan Area Networks–Bridges and Bridged Networks–Amendment 28: Per-Stream Filtering and Policing*, Standard 802.1Qci-2017, 2017, pp. 1–65.

[85] J. Yan, W. Quan, X. Jiang, and Z. Sun, ''Injection time planning: Making CQF practical in time-sensitive networking,'' in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Jul. 2020, pp. 616–625.

[86] J. Zhang, Q. Xu, X. Lu, Y. Zhang, and C. Chen, "Coordinated data transmission in time-sensitive networking for mixed time-sensitive applications," in *Proc. IECON 46th Annu. Conf. IEEE Ind. Electron. Soc.*, Oct. 2020, pp. 3805–3810.

[87] D. Thiele, R. Ernst, and J. Diemer, "Formal worst-case timing analysis of Ethernet TSN's time-aware and peristaltic shapers," in *Proc. IEEE Veh. Netw. Conf. (VNC)*, Dec. 2015, pp. 251–258.

[88] C. Park, J. Lee, T. Tan, and S. Park, "Simulation of scheduled traffic for the IEEE 802.1 time sensitive networking," in *Information Science and Applications (ICISA)*. Cham, Switzerland: Springer, 2016, pp. 75–83.

[89] F. Smirnov, M. Glas, F. Reimann, and J. Teich, "Formal timing analysis of non-scheduled traffic in automotive scheduled TSN networks," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2017, pp. 1643–1646.

[90] A. M. Kentis, M. S. Berger, and J. Soler, "Effects of port congestion in the gate control list scheduling of time sensitive networks," in *Proc. 8th Int. Conf. Netw. Future (NOF)*, Nov. 2017, pp. 138–140.

[91] L. Zhao, P. Pop, and S. S. Craciunas, "Worst-case latency analysis for IEEE 802.1Qbv time sensitive networks using network calculus," *IEEE Access*, vol. 6, pp. 41803–41815, 2018.

[92] T. L. Mai, N. Navet, and J. Migge, "A hybrid machine learning and schedulability analysis method for the verification of TSN networks," in *Proc. 15th IEEE Int. Workshop Factory Commun. Syst. (WFCS)*, May 2019, pp. 1–8.

[93] T. L. Mai, N. Navet, and J. Migge, "On the use of supervised machine learning for assessing schedulability: Application to Ethernet TSN," in *Proc. 27th Int. Conf. Real-Time Netw. Syst. (RTNS)*, 2019, pp. 143–153.

[94] L. Velasco and M. Ruiz, "Supporting time-sensitive and best-effort traffic on a common metro infrastructure," *IEEE Commun. Lett.*, vol. 24, no. 8, pp. 1664–1668, Aug. 2020.

[95] L. Lo Bello, M. Ashjaei, G. Patti, and M. Behnam, "Schedulability analysis of time-sensitive networks with scheduled traffic and preemption support," *J. Parallel Distrib. Comput.*, vol. 144, pp. 153–171, Oct. 2020.

[96] C. Simon, M. Maliosz, and M. Mate, "Design aspects of low-latency services with time-sensitive networking," *IEEE Commun. Standards Mag.*, vol. 2, no. 2, pp. 48–54, Jun. 2018.

[97] F. Ansah, M. A. Abid, and H. de Meer, "Schedulability analysis and GCL computation for time-sensitive networks," in *Proc. IEEE 17th Int. Conf. Ind. Informat. (INDIN)*, Jul. 2019, pp. 926–932.

[98] S. Thangamuthu, N. Concer, P. J. L. Cuijpers, and J. J. Lukkien, "Analysis of Ethernet-switch traffic shapers for in-vehicle networking applications," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, 2015, pp. 55–60.

[99] D. Thiele and R. Ernst, "Formal worst-case timing analysis of Ethernet TSN's burst-limiting shaper," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, 2016, pp. 187–192.

[100] M. Boyer and H. Daigmorte, "Impact on credit freeze before gate closing in CBS and GCL integration into TSN," in *Proc. 27th Int. Conf. Real-Time Netw. Syst. (RTNS)*, 2019, pp. 80–89.

[101] E. Mohammadpour, E. Stai, and J.-Y. Le Boudec, "Improved credit bounds for the credit-based shaper in time-sensitive networking," *IEEE Netw. Lett.*, vol. 1, no. 3, pp. 136–139, Sep. 2019.

[102] B. Fang, Q. Li, H. Hu, and H. Xiong, "Token regulated credit based shaper for time sensitive networks," in *Proc. Int. Conf. Comput., Inf. Telecommun. Syst. (CITS)*, Oct. 2020, pp. 1–6.

[103] F. He, L. Zhao, and E. Li, "Impact analysis of flow shaping in Ethernet-AVB/TSN and AFDX from network calculus and simulation perspective," *Sensors*, vol. 17, no. 5, p. 1181, May 2017.

[104] D. Maxim and Y.-Q. Song, "Delay analysis of AVB traffic in time-sensitive networks (TSN)," in *Proc. 25th Int. Conf. Real-Time Netw. Syst.*, Oct. 2017, pp. 18–27.

[105] L. Zhao, P. Pop, Z. Zheng, and Q. Li, "Timing analysis of AVB traffic in TSN networks using network calculus," in *Proc. IEEE Real-Time Embedded Technol. Appl. Symp. (RTAS)*, Apr. 2018, pp. 25–36.

[106] J. Zhang, L. Chen, T. Wang, and X. Wang, "Analysis of TSN for industrial automation based on network calculus," in *Proc. 24th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2019, pp. 240–247.

[107] J. Specht and S. Samii, "Urgency-based scheduler for time-sensitive switched Ethernet networks," in *Proc. 28th Euromicro Conf. Real-Time Syst. (ECRTS)*, Jul. 2016, pp. 75–85.

[108] J.-Y. Le Boudec, "A theory of traffic regulators for deterministic networks with application to interleaved regulators," *IEEE/ACM Trans. Netw.*, vol. 26, no. 6, pp. 2721–2733, Dec. 2018.

[109] J. Specht and S. Samii, "Synthesis of queue and priority assignment for asynchronous traffic shaping in switched Ethernet," in *Proc. IEEE Real-Time Syst. Symp. (RTSS)*, Dec. 2017, pp. 178–187.

[110] E. Mohammadpour, E. Stai, M. Mohiuddin, and J.-Y. Le Boudec, "Latency and backlog bounds in time-sensitive networking with credit based shapers and asynchronous traffic shaping," in *Proc. 30th Int. Teletraffic Congr. (ITC)*, Sep. 2018, pp. 1–6.

[111] A. Grigorjew, F. Metzger, T. Hossfeld, and J. Specht, "A simulation of asynchronous traffic shapers in switched Ethernet networks," in *Proc. Int. Conf. Netw. Syst. (NetSys)*, Mar. 2019, pp. 1–6.

[112] L. Thomas and J.-Y. Le Boudec, "On time synchronization issues in time-sensitive networks with regulators and nonideal clocks," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 4, no. 2, pp. 1–41, Jun. 2020.

[113] B. Fang, Q. Li, Z. Gong, and H. Xiong, "Simulative assessments of credit-based shaping and asynchronous traffic shaping in time-sensitive networking," in *Proc. 12th Int. Conf. Adv. INFOCOMM Technol. (ICAIT)*, Nov. 2020, pp. 111–118.

[114] A. Nasrallah, A. S. Thyagaturu, Z. Alharbi, C. Wang, X. Shao, M. Reisslein, and H. Elbakoury, "Performance comparison of IEEE 802.1 TSN time aware shaper (TAS) and asynchronous traffic shaper (ATS)," *IEEE Access*, vol. 7, pp. 44165–44181, 2019.

[115] Z. Zhou, Y. Yan, M. Berger, and S. Ruepp, "Analysis and modeling of asynchronous traffic shaping in time sensitive networks," in *Proc. 14th IEEE Int. Workshop Factory Commun. Syst. (WFCS)*, Jun. 2018, pp. 1–4.

[116] Z. Zhou, M. S. Berger, S. R. Ruepp, and Y. Yan, "Insight into the IEEE 802.1 qcr asynchronous traffic shaping in time sensitive network," *Adv. Sci., Technol. Eng. Syst. J.*, vol. 4, no. 1, pp. 292–301, 2019.

[117] Z. Zhou, M. S. Berger, and Y. Yan, "Mapping TSN traffic scheduling and shaping to FPGA-based architecture," *IEEE Access*, vol. 8, pp. 221503–221512, 2020.

[118] J. Prados-Garzon, T. Taleb, and M. Bagaa, "LEARNET: Reinforcement learning based flow scheduling for asynchronous deterministic networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6.

[119] H. Lee, J. Lee, C. Park, and S. Park, "Time-aware preemption to enhance the performance of audio/video bridging (AVB) in IEEE 802.1 TSN," in *Proc. 1st IEEE Int. Conf. Comput. Commun. Internet (ICCCI)*, Oct. 2016, pp. 80–84.

[120] J. Lee and S. Park, "Time-sensitive network (TSN) experiment in sensor-based integrated environment for autonomous driving," *Sensors*, vol. 19, no. 5, p. 1111, Mar. 2019.

[121] Z. Zhou, Y. Yan, S. Ruepp, and M. Berger, "Analysis and implementation of packet preemption for time sensitive networks," in *Proc. IEEE 18th Int. Conf. High Perform. Switching Routing (HPSR)*, Jun. 2017, pp. 1–6.

[122] S. Kehrer, O. Kleineberg, and D. Heffernan, "A comparison of fault-tolerance concepts for IEEE 802.1 time sensitive networks (TSN)," in *Proc. IEEE Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2014, pp. 1–8.

[123] I. Alvarez, J. Proenza, M. Barranco, and M. Knezic, "Towards a time redundancy mechanism for critical frames in time-sensitive networking," in *Proc. 22nd IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2017, pp. 1–4.

[124] I. Alvarez, D. Cavka, J. Proenza, and M. Barranco, "Simulation of the proactive transmission of replicated frames mechanism over TSN," in *Proc. 24th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2019, pp. 1375–1378.

[125] V. Gavrilut, B. Zarrin, P. Pop, and S. Samii, "Fault-tolerant topology and routing synthesis for IEEE time-sensitive networking," in *Proc. 25th Int. Conf. Real-Time Netw. Syst.*, Oct. 2017, pp. 267–276.

[126] F. Smirnov, F. Reimann, J. Teich, Z. Han, and M. Glaß, "Automatic optimization of redundant message routings in automotive networks," in *Proc. 21st Int. Workshop Softw. Compil. Embedded Syst.*, May 2018, pp. 90–99.

[127] A. A. Atallah, G. B. Hamad, and O. A. Mohamed, "Fault-resilient topology planning and traffic configuration for IEEE 802.1Qbv TSN networks," in *Proc. IEEE 24th Int. Symp. Line Test. Robust Syst. Design (IOLTS)*, Jul. 2018, pp. 151–156.

[128] A. A. Atallah, G. B. Hamad, and O. A. Mohamed, "Reliability-aware routing of AVB streams in TSN networks," in *Proc. Int. Conf. Ind., Eng. Other Appl. Appl. Intell. Syst.* Cham, Switzerland: Springer, 2018, pp. 697–708.

[129] R. Dobrin, N. Desai, and S. Punnekkat, "On fault-tolerant scheduling of time sensitive networks," in *Proc. Int. Workshop Secur. Dependability Crit. Embedded Real-Time Syst. (CERTS)*. Wadern, Germany: Schloss Dagstuhl-Leibniz-Zentrum Fuer Informatik, 2019, p. 15.

[130] M. A. Ojewale and P. M. Yomsi, "Routing heuristics for load-balanced transmission in TSN-based networks," *ACM SIGBED Rev.*, vol. 16, no. 4, pp. 20–25, Jan. 2020.

[131] A. A. Atallah, G. B. Hamad, and O. A. Mohamed, "Reliability analysis of TSN networks under SEU induced soft error using model checking," in *Proc. IEEE Latin Amer. Test Symp. (LATS)*, Mar. 2019, pp. 1–6.

[132] O. Daniel, G. E. Juan, C. Lua, and O. Roman, "Failure detection in TSN startup using deep learning," in *Proc. IEEE 23rd Int. Symp. Real-Time Distrib. Comput. (ISORC)*, May 2020, pp. 140–141.

[133] N. Desai and S. Punnekkat, "Enhancing fault detection in time sensitive networks using machine learning," in *Proc. Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Jan. 2020, pp. 714–719.

[134] K. Stanton, "Distributing deterministic, accurate time for tightly coordinated network and software applications: IEEE 802.1 AS, the TSN profile of PTP," *IEEE Commun. Standards Mag.*, vol. 2, no. 2, pp. 34–40, Jun. 2018.

[135] H. Zhu, K. Liu, Y. Yan, H. Zhang, and T. Huang, "Measures to improve the accuracy and reliability of clock synchronization in time-sensitive networking," *IEEE Access*, vol. 8, pp. 192368–192378, 2020.

[136] 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 Medit. Conf. Embedded Comput. (MECO)*, Jun. 2020, pp. 1–8.

[137] D. B. Mateu, D. Hallmans, M. Ashjaei, A. V. Papadopoulos, J. Proenza, and T. Nolte, "Clock synchronization in integrated TSN-EtherCAT networks," in *Proc. 25th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2020, pp. 214–221.

[138] *IEEE Standard for Local and Metropolitan Area Networks–Virtual Bridged Local Area Networks Amendment 14: Stream Reservation Protocol (SRP)*, Standard 802.1Qat-2010, 2010, pp. 1–119.

[139] D. Park, J. Lee, C. Park, and S. Park, "New automatic de-registration method utilizing a timer in the IEEE802.1 TSN," in *Proc. 1st IEEE Int. Conf. Comput. Commun. Internet (ICCCI)*, Oct. 2016, pp. 47–51.

[140] P. Zhang, Y. Liu, J. Shi, Y. Huang, and Y. Zhao, "A feasibility analysis framework of time-sensitive networking using real-time calculus," *IEEE Access*, vol. 7, pp. 90069–90081, 2019.

[141] M. H. Farzaneh, S. Shafaei, and A. Knoll, "Formally verifiable modeling of in-vehicle time-sensitive networks (TSN) based on logic programming," in *Proc. IEEE Veh. Netw. Conf. (VNC)*, Dec. 2016, pp. 1–4.

[142] M. H. Farzaneh and A. Knoll, "An ontology-based plug-and-play approach for in-vehicle time-sensitive networking (TSN)," in *Proc. IEEE 7th Annu. Inf. Technol., Electron. Mobile Commun. Conf. (IEMCON)*, Oct. 2016, pp. 1–8.

[143] S. Mubeen, M. Ashjaei, and M. Sjodin, "Holistic modeling of time sensitive networking in component-based vehicular embedded systems," in *Proc. 45th Euromicro Conf. Softw. Eng. Adv. Appl. (SEAA)*, Aug. 2019, pp. 131–139.

[144] M. H. Farzaneh, S. Kugele, and A. Knoll, "A graphical modeling tool supporting automated schedule synthesis for time-sensitive networking," in *Proc. 22nd IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2017, pp. 1–8.

[145] *IEEE Standard for Local and Metropolitan Area Networks–Bridges and Bridged Networks–Amendment 30: YANG Data Model*, Standard 802.1Qcp-2018, 2018, pp. 1–93.

[146] M. Gutierrez, A. Ademaj, W. Steiner, R. Dobrin, and S. Punnekkat, "Self-configuration of IEEE 802.1 TSN networks," in *Proc. 22nd IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2017, pp. 1–8.

[147] A. Nasrallah, V. Balasubramanian, A. Thyagaturu, M. Reisslein, and H. ElBakoury, "Reconfiguration algorithms for high precision communications in time sensitive networks," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2019, pp. 1–6.

[148] N. G. Nayak, F. Dürr, and K. Rothermel, "Time-sensitive software-defined network (TSSDN) for real-time applications," in *Proc. 24th Int. Conf. Real-Time Netw. Syst. (RTNS)*, 2016, pp. 193–202.

[149] D. Thiele and R. Ernst, "Formal analysis based evaluation of software defined networking for time-sensitive Ethernet," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, 2016, pp. 31–36.

[150] L. Silva, P. Pedreiras, P. Fonseca, and L. Almeida, "On the adequacy of SDN and TSN for industry 4.0," in *Proc. IEEE 22nd Int. Symp. Real-Time Distrib. Comput. (ISORC)*, May 2019, pp. 43–51.

[151] F. Prinz, M. Schoeffler, A. Lechler, and A. Verl, "Dynamic real-time orchestration of I4.0 components based on time-sensitive networking," *Proc. CIRP*, vol. 72, pp. 910–915, Jan. 2018.

[152] M. A. Metaal, R. Guillaume, R. Steinmetz, and A. Rizk, "Integrated industrial Ethernet networks: Time-sensitive networking over SDN infrastructure for mixed applications," in *Proc. IEEE IFIP Netw. Conf. Netw.*, Jun. 2020, pp. 803–808.

[153] M. L. Raagaard, P. Pop, M. Gutierrez, and W. Steiner, "Runtime reconfiguration of time-sensitive networking (TSN) schedules for fog computing," in *Proc. IEEE Fog World Congr. (FWC)*, Oct. 2017, pp. 1–6.

[154] P. Pop, M. L. Raagaard, M. Gutierrez, and W. Steiner, "Enabling fog computing for industrial automation through time-sensitive networking (TSN)," *IEEE Commun. Standards Mag.*, vol. 2, no. 2, pp. 55–61, Jun. 2018.

[155] *OPC Foundation, Open Platform Communication Unified Architecture*. Accessed: Jul. 2021. [Online]. Available: https://opcfoundation.org/about/opc-technologies/opc-ua/

[156] D. Bruckner, M.-P. Stanica, R. Blair, S. Schriegel, S. Kehrer, M. Seewald, and T. Sauter, "An introduction to OPC UA TSN for industrial communication systems," *Proc. IEEE*, vol. 107, no. 6, pp. 1121–1131, Jun. 2019.

[157] S. Tian and Y. Hu, "The role of OPC UA TSN in IT and OT convergence," in *Proc. Chin. Autom. Congr. (CAC)*, Nov. 2019, pp. 2272–2276.

[158] F. Palm, S. Gruner, J. Pfrommer, M. Graube, and L. Urbas, "Open source as enabler for OPC UA in industrial automation," in *Proc. IEEE 20th Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2015, pp. 1–6.

[159] L. Sciullo, S. Bhattacharjee, and M. Kovatsch, "Bringing deterministic industrial networking to the W3C web of things with TSN and OPC UA," in *Proc. 10th Int. Conf. Internet Things*, Oct. 2020, pp. 1–8.

[160] M. H. Farzaneh and A. Knoll, "Time-sensitive networking (TSN): An experimental setup," in *Proc. IEEE Veh. Netw. Conf. (VNC)*, Nov. 2017, pp. 23–26.

[161] S. Brunner, J. Roder, M. Kucera, and T. Waas, "Automotive E/E-architecture enhancements by usage of Ethernet TSN," in *Proc. 13th Workshop Intell. Solutions Embedded Syst. (WISES)*, Jun. 2017, pp. 9–13.

[162] S. Schlichting, A. Drägerwerk, C. KGaA, and S. Pöhlsen, "An architecture for distributed systems of medical devices in high acuity environments," Dräger, Lübeck, Germany, Tech. Rep., 2014. [Online]. Available: https://www.hl7.org/documentcenter/public/wg/healthcaredevices/20140116%20An%20architecture%20for%20distributed%20systems%20of%20medical%20devices%20in%20high-acuity%20environments.pdf

[163] *openSAFETY, Introduction*. Accessed: Jul. 2021. [Online]. Available: https://www.ethernet-powerlink.org/open-safety/introduction

[164] B. Rother, M. Kasparick, E. Schweisguth, F. Golatowski, and D. Timmermann, "Automatic configuration of a TSN network for SDC-based medical device networks," in *Proc. 16th IEEE Int. Conf. Factory Commun. Syst. (WFCS)*, Apr. 2020, pp. 1–8.

[165] S. Gent, P. G. Peon, T. Fruhwirth, and D. Etz, "Hosting functional safety applications in factory networks through time-sensitive networking," in *Proc. 25th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2020, pp. 230–237.

[166] J. Sanchez-Garrido, A. Jurado, L. Medina, R. Rodriguez, E. Ros, and J. Diaz, "Digital electrical substation communications based on deterministic time-sensitive networking over Ethernet," *IEEE Access*, vol. 8, pp. 93621–93634, 2020.

[167] R. E. Mackiewicz, "Overview of IEC 61850 and benefits," in *Proc. IEEE Power Eng. Soc. Gen. Meeting*, Jun. 2006, p. 8.

[168] T. Docquier, Y.-Q. Song, V. Chevrier, L. Pontnau, and A. Ahmed-Nacer, "IEC 61850 over TSN: Traffic mapping and delay analysis of GOOSE traffic," in *Proc. 25th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2020, pp. 246–253.

[169] F. Golatowski, H. Raddatz, F. Holzke, E. Janchivnyambuu, P. Danielis, and D. Timmermann, "An OPC UA-based crane model using time-sensitive networking technology," in *Proc. IEEE Conf. Ind. Cyberphys. Syst. (ICPS)*, Jun. 2020, pp. 537–542.

[170] T. Agarwal, P. Niknejad, M. R. Barzegaran, and L. Vanfretti, "Multi-level time-sensitive networking (TSN) using the data distribution services (DDS) for synchronized three-phase measurement data transfer," *IEEE Access*, vol. 7, pp. 131407–131417, 2019.

[171] J. Yang, K. Sandstrom, T. Nolte, and M. Behnam, "Data distribution service for industrial automation," in *Proc. IEEE 17th Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2012, pp. 1–8.

[172] D. Cavalcanti, J. Perez-Ramirez, M. M. Rashid, J. Fang, M. Galeev, and K. B. Stanton, "Extending accurate time distribution and timeliness capabilities over the air to enable future wireless industrial automation systems," *Proc. IEEE*, vol. 107, no. 6, pp. 1132–1152, Jun. 2019.

[173] O. Seijo, J. A. Lopez-Fernandez, H.-P. Bernhard, and I. Val, "Enhanced timestamping method for subnanosecond time synchronization in IEEE 802.11 over WLAN standard conditions," *IEEE Trans. Ind. Informat.*, vol. 16, no. 9, pp. 5792–5805, Sep. 2020.

[174] D. Shrestha, Z. Pang, and D. Dzung, "Precise clock synchronization in high performance wireless communication for time sensitive networking," *IEEE Access*, vol. 6, pp. 8944–8953, 2018.

[175] A. M. Romanov, F. Gringoli, and A. Sikora, "A precise synchronization method for future wireless TSN networks," *IEEE Trans. Ind. Informat.*, vol. 17, no. 5, pp. 3682–3692, May 2021.

[176] E. Genc and L. F. Del Carpio, "Wi-Fi QoS enhancements for downlink operations in industrial automation using TSN," in *Proc. 15th IEEE Int. Workshop Factory Commun. Syst. (WFCS)*, May 2019, pp. 1–6.

[177] A. Neumann, L. Wisniewski, R. S. Ganesan, P. Rost, and J. Jasperneite, "Towards integration of industrial Ethernet with 5G mobile networks," in *Proc. 14th IEEE Int. Workshop Factory Commun. Syst. (WFCS)*, Jun. 2018, pp. 1–4.

[178] M. Khoshnevisan, V. Joseph, P. Gupta, F. Meshkati, R. Prakash, and P. Tinnakornsrisuphap, "5G industrial networks with comp for URLLC and time sensitive network architecture," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 4, pp. 947–959, Apr. 2019.

[179] D. Ginthor, J. von Hoyningen-Huene, R. Guillaume, and H. Schotten, "Analysis of multi-user scheduling in a TSN-enabled 5G system for industrial applications," in *Proc. IEEE Int. Conf. Ind. Internet (ICII)*, Nov. 2019, pp. 190–199.

[180] A. Larranaga, M. C. Lucas-Estan, I. Martinez, I. Val, and J. Gozalvez, "Analysis of 5G-TSN integration to support industry 4.0," in *Proc. 25th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2020, pp. 1111–1114.

[181] D. Ginthor, R. Guillaume, J. von Hoyningen-Huene, M. Schungel, and H. D. Schotten, "End-to-end optimized joint scheduling of converged wireless and wired time-sensitive networks," in *Proc. 25th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2020, pp. 222–229.

[182] I. Godor, M. Luvisotto, S. Ruffini, K. Wang, D. Patel, J. Sachs, O. Dobrijevic, D. P. Venmani, O. L. Moult, J. Costa-Requena, A. Poutanen, C. Marshall, and J. Farkas, "A look inside 5G standards to support time synchronization for smart manufacturing," *IEEE Commun. Standards Mag.*, vol. 4, no. 3, pp. 14–21, Sep. 2020.

[183] *IEEE Standard for Local and Metropolitan Area Networks–Time-Sensitive Networking for Fronthaul*, Standard 802.1CM-2018, 2018, pp. 1–62.

[184] S. Bhattacharjee, R. Schmidt, K. Katsalis, C.-Y. Chang, T. Bauschert, and N. Nikaein, "Time-sensitive networking for 5G fronthaul networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–7.

[185] Y. Liu, Y. Zhou, J. Yuan, and L. Liu, "Delay aware flow scheduling for time sensitive fronthaul networks in centralized radio access network," *IEEE Trans. Commun.*, vol. 68, no. 5, pp. 2992–3009, May 2020.

[186] S. Zhang, X. Gong, P. Wang, L. Wang, T. Wang, X. Que, and Y. Tian, "Deterministic transmittable time-based asynchronous scheduler for fronthaul networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2019, pp. 1–6.

[187] F. Ansah, J. Rottmeier, A. Zirkler, and H. de Meer, "Worst-case delay slicing for time-sensitive applications in softwarized industrial networks," in *Proc. 25th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2020, pp. 1652–1659.

[188] P. Heise, F. Geyer, and R. Obermaisser, "TSimNet: An industrial time sensitive networking simulation framework based on OMNeT++," in *Proc. 8th IFIP Int. Conf. New Technol., Mobility Secur. (NTMS)*, Nov. 2016, pp. 1–5.

[189] J. Jiang, Y. Li, S. H. Hong, A. Xu, and K. Wang, "A time-sensitive networking (TSN) simulation model based on OMNET++," in *Proc. IEEE Int. Conf. Mechatronics Autom. (ICMA)*, Aug. 2018, pp. 643–648.

[190] *Communication Over Real-Time Ethernet Group*. Accessed: Jul. 2021. [Online]. Available: https://core-researchgroup.de/

[191] L. Martenvormfelde, A. Neumann, L. Wisniewski, and J. Jasperneite, "A simulation model for integrating 5G into time sensitive networking as a transparent bridge," in *Proc. 25th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2020, pp. 1103–1106.

[192] M. Pahlevan, B. Balakrishna, and R. Obermaisser, "Simulation framework for clock synchronization in time sensitive networking," in *Proc. IEEE 22nd Int. Symp. Real-Time Distrib. Comput. (ISORC)*, May 2019, pp. 213–220.

[193] M. K. Al-Hares, P. Assimakopoulos, D. Muench, and N. J. Gomes, "Modeling time aware shaping in an Ethernet fronthaul," in *Proc. GLOBECOM IEEE Global Commun. Conf.*, Dec. 2017, pp. 1–6.

[194] M. Pahlevan and R. Obermaisser, "Evaluation of time-triggered traffic in time-sensitive networks using the OPNET simulation framework," in *Proc. 26th Euromicro Int. Conf. Parallel, Distrib. Netw.-Based Process. (PDP)*, Mar. 2018, pp. 283–287.

[195] M. Pahlevan and R. Obermaisser, "Redundancy management for safety-critical applications with time sensitive networking," in *Proc. 28th Int. Telecommun. Netw. Appl. Conf. (ITNAC)*, Nov. 2018, pp. 1–7.

[196] G. Behrmann, A. David, and K. G. Larsen, "A tutorial on UPPAAL," in *Formal Methods for the Design of Real-Time Systems* (Lecture Notes in Computer Science), vol. 3185, Berlin, Germany: Springer, 2004, pp. 200–236.

[197] J. Lv, Y. Zhao, X. Wu, Y. Li, and Q. Wang, "Formal analysis of TSN scheduler for real-time communications," *IEEE Trans. Rel.*, vol. 70, no. 3, pp. 1286–1294, Sep. 2021.

[198] E. Kyriakakis, J. Sparsø, and M. Schoeberl, "Hardware assisted clock synchronization with the IEEE 1588-2008 precision time protocol," in *Proc. 26th Int. Conf. Real-Time Netw. Syst.*, Oct. 2018, pp. 51–60.

[199] F. Gross, T. Steinbach, F. Korf, T. C. Schmidt, and B. Schwarz, "A hardware/software co-design approach for Ethernet controllers to support time-triggered traffic in the upcoming IEEE TSN standards," in *Proc. IEEE 4th Int. Conf. Consum. Electron. Berlin (ICCE-Berlin)*, Sep. 2014, pp. 9–13.

[200] J. Coleman, S. Almalih, A. Slota, and Y.-H. Lee, "Emerging COTS architecture support for real-time TSN Ethernet," in *Proc. 34th ACM/SIGAPP Symp. Appl. Comput.*, Apr. 2019, pp. 258–265.

[201] A. Pruski and M. Berger, "Design considerations for high-performance time sensitive networking switches," in *Proc. 10th Int. Conf. Netw. Future (NoF)*, Oct. 2019, pp. 114–117.

[202] Z. Li, H. Wan, Y. Deng, X. Zhao, Y. Gao, X. Song, and M. Gu, "Time-triggered switch-memory-switch architecture for time-sensitive networking switches," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 1, pp. 185–198, Jan. 2020.

[203] L. Maile, K.-S. Hielscher, and R. German, "Network calculus results for TSN: An introduction," in *Proc. Inf. Commun. Technol. Conf. (ICTC)*, May 2020, pp. 131–140.

[204] S. Samii and H. Zinner, "Level 5 by layer 2: Time-sensitive networking for autonomous vehicles," *IEEE Commun. Standards Mag.*, vol. 2, no. 2, pp. 62–68, Jun. 2018.

[205] *SAE International, Taxonomy and Definitions for Terms Related to Driving Automation Systems for on-Road Motor Vehicles*. Accessed: Jul. 2021. [Online]. Available: https://www.sae.org/standards/content/j3016_202104/

[206] D. Hallmans, M. Ashjaei, and T. Nolte, "Analysis of the TSN standards for utilization in long-life industrial distributed control systems," in *Proc. 25th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2020, pp. 190–197.

[207] *IEEE Standard for Local and Metropolitan Area Networks–Bridges and Bridged Networks–Amendment 24: Path Control and Reservation*, Standard 802.1Qca-2016, 2016, pp. 1–120.

YOUHWAN SEOL received the B.S. degree from the School of Computer Science and Engineering, Chung-Ang University, Seoul, Republic of Korea, in 2019, where he is currently pursuing the M.S. degree with the Department of Computer Science and Engineering. He is also a Research Assistant with the Networked Systems Laboratory (NSL), Chung-Ang University, led by Dr. J. Paek, with research interests in blockchain and time-sensitive networking.

DOYEON HYEON received the B.S. degree from the School of Integrative Engineering, Chung-Ang University, Seoul, Republic of Korea, in 2020, where she is currently pursuing the M.S. degree with the Department of Computer Science and Engineering. She is also a Research Assistant with the Networked Systems Laboratory (NSL), Chung-Ang University, led by Dr. J. Paek, with research interests in time sensitive networking.

JUNHONG MIN received the B.S. degree from the School of Computer Science and Engineering, Chung-Ang University, Seoul, Republic of Korea, in 2021, where he is currently pursuing the M.S. degree with the Department of Computer Science and Engineering. He is also a Research Assistant with the Networked Systems Laboratory (NSL), Chung-Ang University, led by Dr. J. Paek, with research interests in machine learning and time-sensitive networking.

MOONBEOM KIM received the B.S. degree in computer and information communications engineering from Hongik University, in 2017, and the M.S. degree in computer science and engineering from Chung-Ang University, Seoul, Republic of Korea, in 2020, where he is currently pursuing the Ph.D. degree in computer science and engineering. He is also a research assistant with the Networked Systems Laboratory (NSL), Chung-Ang University, led by Dr. J. Paek, with research interests in wireless networking, localization, and time-sensitive networking.

JEONGYEUP PAEK (Senior Member, IEEE) received the B.S. degree from Seoul National University, in 2003, and the M.S. and Ph.D. degrees from the University of Southern California (USC), in 2005 and 2010, respectively. He worked at Deutsche Telekom R&D Labs, USA, as a Research Intern, in 2010, and then, he joined Cisco Systems, in 2011, where he was a Technical Leader with the Internet of Things Group (IoTG), Connected Energy Networks Business Unit (CENBU, formerly Smart Grid BU). In 2014, he was with the Department of Computer Information Communication, Hongik University, as an Assistant Professor.