Time-Sensitive Networking (TSN) for Industrial Automation: Current Advances and Future Directions

TIANYU ZHANG, Computer Science, The University of Iowa, Iowa City, United States
GANG WANG, Emerson Automation Solutions, Round Rock, United States
CHUANYU XUE, University of Connecticut, Storrs, United States
JIACHEN WANG, University of Connecticut, Storrs, United States
MARK NIXON, Emerson Automation Solutions, Round Rock, United States
SONG HAN, University of Connecticut, Storrs, United States

With the introduction of Cyber-Physical Systems (CPS) and Internet of Things (IoT) technologies, the automation industry is undergoing significant changes, particularly in improving production efficiency and reducing maintenance costs. Industrial automation applications often need to transmit time- and safety-critical data to closely monitor and control industrial processes. Several Ethernet-based fieldbus solutions, such as PROFINET IRT, EtherNet/IP, and EtherCAT, are widely used to ensure real-time communications in industrial automation systems. These solutions, however, commonly incorporate additional mechanisms to provide latency guarantees, making their interoperability a grand challenge. The IEEE 802.1 Time-Sensitive Networking (TSN) task group was formed to enhance and optimize IEEE 802.1 network standards, particularly for Ethernet-based networks. These solutions can be evolved and adapted for cross-industry scenarios, such as large-scale distributed industrial plants requiring multiple industrial entities to work collaboratively. This paper provides a comprehensive review of current advances in TSN standards for industrial automation. It presents the state-of-the-art IEEE TSN standards and discusses the opportunities and challenges of integrating TSN into the automation industry. Some promising research directions are also highlighted for applying TSN technologies to industrial automation applications.

CCS Concepts: • **General and reference** → **Surveys and overviews**; • **Networks** → **Packet-switching networks**; **Link-layer protocols;**

Additional Key Words and Phrases: Time-Sensitive Networking (TSN), IEEE 802.1Q, industrial automation

ACM Reference Format:
Tianyu Zhang, Gang Wang, Chuanyu Xue, Jiachen Wang, Mark Nixon, and Song Han. 2024. Time-Sensitive Networking (TSN) for Industrial Automation: Current Advances and Future Directions. *ACM Comput. Surv.* 57, 2, Article 30 (October 2024), 38 pages. https://doi.org/10.1145/3695248

T. Zhang and G. Wang contributed equally to this paper. Part of this work was conducted while Tianyu Zhang was affiliated with the University of Connecticut.

The work is supported in part by the National Science Foundation Grant CNS-1932480, CNS-2008463, CCF-2028875, CNS-1925706, and the NASA STRI Resilient Extraterrestrial Habitats Institute (RETHi) under grant number 80NSSC19K1076.

Authors' Contact Information: Tianyu Zhang, Computer Science, The University of Iowa, Iowa City, Iowa, United States; e-mail: tianyu-zhang@uiowa.edu; Gang Wang, Emerson Automation Solutions, Round Rock, Texas, United States; e-mail: gang.wang2@emerson.com; Chuanyu Xue, University of Connecticut, Storrs, Connecticut, United States; e-mail: chuanyu.xue@uconn.edu; Jiachen Wang, University of Connecticut, Storrs, Connecticut, United States; e-mail: jc.wang@uconn.edu; Mark Nixon, Emerson Automation Solutions, Round Rock, Texas, United States; e-mail: mark.nixon@emerson.com; Song Han, University of Connecticut, Storrs, Connecticut, United States; e-mail: song.han@uconn.edu.

ACM 0360-0300/2024/10-ART30
https://doi.org/10.1145/3695248

1 Introduction

Industrial automation systems commonly employ a hierarchical architecture to perform designed control and automation processes [81]. Ethernet-based fieldbus communication systems are currently dominating the automation industry, with multiple protocols and standards available [128]. However, different vendors may select different industrial Ethernet protocols for use in their devices, resulting in incompatibilities among the deployed equipment. This phenomenon contributes to industrial automation architectures being hierarchical, custom-built, and inflexible when integrating devices from different vendors or standards [69]. Fortunately, driven by the recent advances in Industrial **Internet of Things (IIoT)** technologies, many technical initiatives are pushing industrial automation applications to be more flexible, interoperable, and seamless. One of the most important requirements for industrial automation is real-time and deterministic communication, which is essential for realizing mission-critical control processes [25].

Critical traffic flows generated by industrial automation applications require bounded low latency and low jitter to improve production efficiency and reduce communication costs. Typically, these critical traffic flows need to share the communication medium (e.g., Ethernet) with non-critical flows (e.g., those with less severe timing constraints) originating from the same applications. Under these conditions, it is imperative to guarantee the timing behavior of critical traffic and provide temporal isolation from non-critical communications. The IEEE 802.1 **Time-Sensitive Networking Task Group (TSN TG)**, evolved from the former IEEE 802.1 **Audio Video Bridging (AVB)** TG, addresses this need by designing general-purpose protocols applicable to various fields, such as factory automation, process automation, substation control, and aerospace applications.

The IEEE TSN TG currently aims to improve the reliability and real-time capabilities of the Ethernet standard (e.g., IEEE 802.3 and IEEE 802.1D). It focuses on several essential aspects of the IEEE AVB standards crucial for industrial automation, including reduced latency, deterministic transmission, independence from physical transmission rates, fault tolerance without additional hardware, and interoperability of solutions from different vendors. Compared to traditional Ethernet-based fieldbus systems, the advantage of TSN is also manifold, including vendor neutrality, higher throughput, more network configuration flexibility, and better scalability [98].

TSN is a collection of standards, standard amendments, and projects published or under development by the TSN TG within the IEEE 802.1 **Working Group (WG)**. There are four main pillars on which TSN is built: (1) time synchronization, (2) guaranteed **end-to-end (e2e)** latency, (3) reliability, and (4) resource management. These characteristics make TSN a strong candidate for meeting special requirements in industrial automation, such as deterministic communication, ultra-low communication latency, and extremely high reliability. While TSN standardization efforts are ongoing, several manufacturers have already demonstrated the promising performance of TSN, showing much higher determinism than current state-of-the-art solutions [10, 82]. However, the benefits of TSN come with challenges that need to be addressed in the deployment of industrial automation systems. These challenges include stringent requirements on network synchronization precision, increased traffic scheduling complexity, integration with wireless devices, and so on.

This paper provides a comprehensive review of the current advances in standardization and research efforts related to TSN for industrial automation. We first give a systematical introduction to the published TSN standards relevant to industrial automation systems and explore the challenges each standard attempts to address. We then highlight how and to what extent these standardization efforts empower Ethernet applications, supporting the new requirements raised by current and future industrial use cases. Note that, in addition to the automation industry, deploying TSN technologies is of great interest in many other industries requiring deterministic, low-latency, and high-reliability communications, including automotive applications [11], aerospace [50], and healthcare [76], which are not the focus of this survey.

The rest of this article is organized as follows. Section 2 provides the background of industrial automation and IEEE TSN technologies. Section 3 describes the up-to-date TSN standardization efforts in detail, and Section 4 discusses the integration of TSN into industrial automation systems. Section 5 discusses the challenges in each category of TSN standards. Section 6 presents the future directions related to TSN R&D, and Section 7 concludes the article.

2 Background

With the introduction of CPS and IoT technologies, the automation industry is undergoing tremendous changes in architecture design and system development. These recent technological advancements enable the interconnection of industrial assets on a broader and more fine-grained scale [148]. In this section, we provide background information on industrial automation and TSN technologies.

2.1 Industrial Automation

Industrial automation is an industry concept that utilizes various sensors, actuators, robotic devices, control systems, and **information technology (IT)** systems to connect and manage different processes and machinery across multiple industries, replacing operations originally performed by humans [15].

2.1.1 Recent Trends in Industrial Automation. The industry has undergone three revolutions: mechanization, electrification, and information. The fourth industrial revolution (also referred to as "Industry 4.0"), currently underway, is marked by the pervasive deployment of IoT devices and services. In this revolution, a wide range of devices are being deployed in a self-organizing manner, typically relying on control and communication systems to manage their operation and interaction. For example, in **Supervisory Control and Data Acquisition (SCADA)** systems [23], proprietary communication systems have been mostly replaced by Sensorbus and fieldbus systems.

The Industry 4.0 revolution posts significantly different requirements on industrial automation systems design. For example, the **Industrial IoT (IIoT)** paradigm advocates for a flat cloud of interconnected devices rather than a complex hierarchy. This shift necessitates a more unified communication system based on IP across all functional layers, where typical requirements on industrial automation systems such as time synchronization, low latency, determinism, and convergence must be met [27]. A flatter hierarchy also demands robust communication systems that support the coexistence of **information technology (IT)** and **operational technology (OT)** systems in industrial automation. Figure 1 illustrates an example of the industrial automation control hierarchy comprising IT and OT components, where IT technologies focus on network connectivity and data communication, whereas OT technologies focus on process operation and the control of field devices [24]. The infrastructure layer provides various transport-oriented protocols to interconnect different IT and OT components.

Industrial automation encompasses a variety of systems, including continuous condition monitoring systems, industrial control systems, and prevention/protection systems. While the functional requirements for different automation systems may vary across domains, they share similarities in terms of physical and logical organization complexity. Additionally, they share common requirements for determinism, reliability, interoperability, and traffic convergence.

— Timing and determinism: Industrial automation typically runs real-time applications with stringent requirements on their temporal behavior and accuracy when responding to internal and external events [117]. Beyond network throughput, the commonly used performance metrics, packet transmission latency, and its time variations (jitter) are critical concerns for many industrial control systems [163]. Timing interactions can complicate

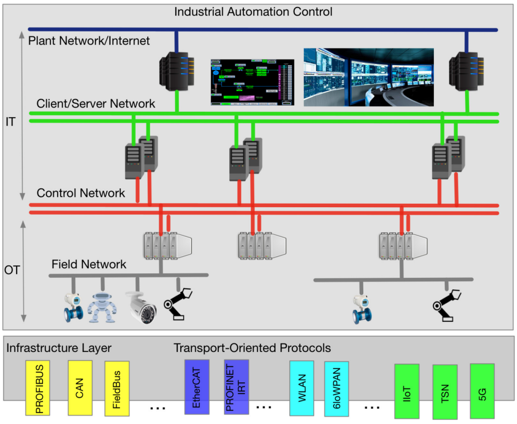

Fig. 1. Example of industrial automation control hierarchy which consists of IT and OT parts.

different procedures. For example, in a switched Ethernet network, achieving deterministic delay is challenging due to the presence of skew or drift in timing signal frames. In addition, the transmission of Ethernet frames can be delayed if the output port on a switching device is busy. These factors accumulate non-deterministic delays in data transmission, which are unsuitable for real-time industrial applications. Therefore, to ensure correct operation, industrial automation systems require a certain degree of determinism.

— Reliability and availability: Production losses in industrial automation due to unexpected stops caused by failure or deterioration of the communication environment are unacceptable. Thus, the reliability and availability[1] of the system are critically important due to the need for accurate and continuous operation in any condition. Reliability can be quantified using appropriate measures such as mean time between failures or the probability of no failure within a specified period of time [56]. Many mission-critical industrial applications often aim for an uptime on the order of 99.999% (known as “five nines” reliability), e.g., 99.9% to 99.9999% for closed-loop control [29].

— Interoperability: An industrial automation system typically consists of diverse devices interconnected through varied technologies. This heterogeneous system architecture necessitates the ability of disparate systems to communicate and share information or resources with one another, known as interoperability. Interoperability is crucial for industrial automation due to its many advantages. For example. by enabling seamless communication and coordination between various systems, businesses can experience enhanced accuracy and productivity. Real-time data exchange and coordinated control across the entire automation system also facilitate efficient decision-making, reducing errors and delays. Interoperability also improves scalability and flexibility, allowing for easier system expansion and modification [28].

— Traffic convergence: Industrial automation applications make use of different traffic types for different functionalities, e.g., sensing, control, alarming, and the like. The diverse traffic types have different characteristics and thus impose varied QoS requirements. The traffic can generally be classified into *critical traffic* and *best-effort* traffic. Critical traffic typically has stringent QoS requirements, and different types of critical traffic may have particular QoS demands depending on the specific application scenarios. IEC/IEEE 60802 group

[1] Reliability and availability are two similar concepts in the context of industrial automation with slight differences. Availability not only takes the possibility of failure but also the possibility of repair into account [18].

summarizes the traffic types for industrial automation (see Table 1 in [77]). Characteristics of these traffic types include deadline and latency, synchronization, transmission period, data size, and interference tolerance. For example, isochronous control loops must meet guaranteed deadline requirements (< 2 *ms*) and cannot tolerate packet loss. While cyclic traffic has more relaxed latency requirements (2 - 20 *ms*) and can tolerate some packet loss (1-4 frames) [6]. In contrast to critical traffic, best-effort traffic generally does not have specific QoS requirements in any of these aspects.

To sum up, industrial automation applications have stringent and specific needs that revolve around ensuring real-time and deterministic communication, high reliability and availability, and interoperability to ensure the efficient, reliable, and safe operation of manufacturing processes and control systems while supporting diverse traffic types. Among these needs, the requirement for deterministic real-time communication, typically evaluated using latency and jitter, plays the most critical role in industrial applications, which we will discuss further below.

2.1.2 Deterministic Real-Time Communication. Packet latency typically refers to an end-to-end (e2e) packet delay from the moment when the sender initiates the transmission to its complete reception by the receiver. The requirement for low latency generally implies that the transmission time must be very short, often within milliseconds, to meet the necessary QoS requirements. Additionally, low-latency applications usually demand deterministic latency. For instance, to ensure the proper functioning of industrial automation systems, all frames within a specified application traffic flow must adhere to a pre-defined latency bound. Some industrial applications also require probabilistic latency. For example, a pre-defined delay bound should be met with high probability, such as in multimedia streaming systems [63], where occasional delay bound violations have negligible effect on perceived multimedia quality.

Latency jitter, or jitter for short, refers to variations in packet latency. Industrial automation systems typically require very low jitter to ensure highly predictable and reliable communication, which is crucial for the proper functioning of industrial processes. Minimizing jitter is essential for maintaining the synchronization and timing precision needed for industrial applications, particularly in motion control, where low jitter is critical for controlling actuation devices. Other industrial applications with low jitter requirements include, but are not limited to, machine tools (100 *ns*), automotive radar (20 *ns*), and professional audio (10 *ns*) [125].

Latency and jitter are the primary QoS metrics for industrial automation. When both packet latency and jitter can be bounded, the communication is considered deterministic, meaning that the message will be transmitted within a specified and predictable time frame. Determinism ensures that communication or output will not only be correct but also occur within a defined period. Industrial automation networks are typically deterministic, catering to many applications requiring such services, including condition monitoring, process automation, and smart manufacturing [117].

2.1.3 The Future of Industrial Ethernet. Currently, Ethernet-based fieldbus systems are prevalent for industrial automation using the widespread Ethernet technology. The implementation of Ethernet to connect field devices offers significant advantages as Ethernet allows for consistent integration at all levels of the hierarchy. In particular, Ethernet enables the vertical and horizontal integration of the industrial automation system from the field level to the application level, which is essential for realizing the vision of IIoT. To achieve the required higher quality of data transmission, **Real-Time Ethernet (RTE)** has become a standard in the automation industry today. However, there is no single standard at present but many different mutually incompatible implementations. Existing RTE solutions can generally be organized into three classes [148].

- Class A: Real-time services with cycle times ranging from 100 *ms*. Example implementations include Modbus-**Interface for Distributed Automation (IDA)**, **Ethernet/Industrial Protocol (IP)**, and **Foundation Fieldbus (FF)** high-speed Ethernet. This class builds on the entire TCP/IP transportation control suite and uses best-effort bridging.
- Class B: Real-time services are performed directly at the top of the **Media Access Control (MAC)** layer using approaches such as prioritization and **Virtual Local Area Network (VLAN)** targeting to separate real-time traffic from the best-effort traffic. For example, using Fast Ethernet [135], the achievable cycle time is within 10 *ms*.
- Class C: Real-time communication is achieved by modifications of the Ethernet MAC layer, including strict traffic scheduling and high-precision clock synchronization. The achievable cycle time can be less than 1 *ms*. Some examples of implementations are EtherCAT [100], **Time-Triggered Ethernet (TTE)** [120] and its variation, **Flexible Time-Triggered Ethernet (FTTE)** [43].

Class C is the most potent class for meeting industrial automation requirements, particularly for TTE and FTTE, which enable determinism in the bandwidth and latency of Ethernet. However, these standards have distinct differences in their support for traffic heterogeneity, time-schedule traffic, time synchronization, and adherence to open standards, thus catering to slightly different needs and markets.[2] Furthermore, as pointed out by [37], which summarizes the requirements of industrial applications into R1 - R7, no industry-established Ethernet-based fieldbus technology can meet all these requirements. Some quantitative performance comparison results among several real-time Ethernet protocols can be found in [26, 106].

Meanwhile, standard Ethernet is evolving towards a real-time communication system that can be applied in industrial applications. The IEEE TSN **Task Group (TG)** is working on improving the reliability and real-time capabilities of Ethernet standards. Specially, the task group addresses several critical shortcomings of the AVB standard, which are vital for industrial automation. These improvements include decreased latency and precise determinism, independence from physical transmission rates, fault tolerance without additional equipment, higher safety and security support, and interoperability among products. In the following sections, we will detail each of these aspects.

2.2 Time-Sensitive Networking (TSN)

TSN offers several advantages to automation industries that have struggled for years with various incompatible proprietary communication protocols. Specifically, TSN ensures vendor-independent interoperability for all features of an industrial system. TSN also addresses scalability issues since it is based on Ethernet, which is highly scalable in end stations and switches.[3] In addition, TSN provides higher flexibility through its standardized technology, enabling the network structure to be flexibly extended without compatibility issues. As an open IEEE standard, TSN can not only ensure seamless communication between devices from different manufacturers but also be integrated with other technologies in the higher layers of the OSI model, such as **OPC Unified Architecture (OPC UA)**, another open and vendor-independent standard. These properties allow for greater interoperability, scalability, and flexibility in industrial automation systems.

To realize the many features TSN provides, the design of a TSN switch plays a fundamental role in making traditional Ethernet have real-time characteristics. A TSN switch is built on a gate driver mechanism and consists of multiple queues per port to buffer traffic with different priorities.

[2] Further discussion and comparison of TTE, FTTE, and TSN can be referred to [42, 170]. This paper primarily focuses on TSN for industrial automation.

[3] In this paper, we use the terms “switch” and “bridge” interchangeably unless otherwise specified.

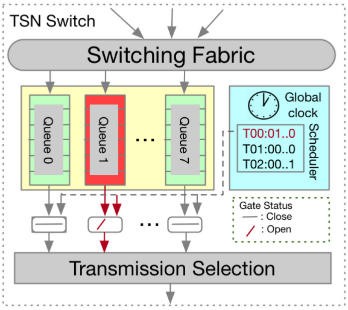

Fig. 2. An illustration of a TSN switch consisting of multiple queues (each equipped with a gate).

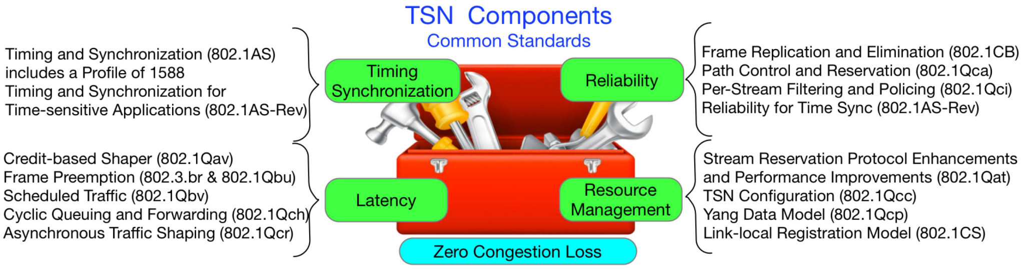
TSN Components
Common Standards
Timing and Synchronization (802.1AS) includes a Profile of 1588
Timing and Synchronization for Time-sensitive Applications (802.1AS-Rev)
Timing Synchronization
Reliability
Frame Replication and Elimination (802.1CB)
Path Control and Reservation (802.1Qca)
Per-Stream Filtering and Policing (802.1Qci)
Reliability for Time Sync (802.1AS-Rev)
Credit-based Shaper (802.1Qav)
Frame Preemption (802.3.br & 802.1Qbu)
Scheduled Traffic (802.1Qbv)
Cyclic Queuing and Forwarding (802.1Qch)
Asynchronous Traffic Shaping (802.1Qcr)
Latency
Resource Management
Stream Reservation Protocol Enhancements and Performance Improvements (802.1Qat)
TSN Configuration (802.1Qcc)
Yang Data Model (802.1Qcp)
Link-local Registration Model (802.1CS)
Zero Congestion Loss

Fig. 3. IEEE 802.1 TSN toolbox, consisting of four coarse subclasses. Some draft standards (e.g., IEEE P802.1ASdm and IEEE P802.1Qdj) are still in progress and thus are not included in the discussion in Section 5.

The forwarded traffic is scheduled according to the control of each gate by carefully determining the time of its opening/closing. Such a mechanism guarantees that the communication delay is predictable and can be managed in a deterministic way. Figure 2 shows an abstract of the TSN switch. It consists of four key components: the switching fabric to filter the traffic, the queues (each equipped with a gate) to buffer the traffic, a global scheduler, and the transmission selection.

Based on the gate driver switch architecture, TSN defines a collection of standards and amendments to meet the demands of industrial automation, especially the deterministic communication of critical traffic in the converged networks. At the highest level, by resource reservation and applying various queuing and shaping technologies, TSN achieves zero congestion loss for critical traffic, and this, in turn, allows a guarantee on the e2e latency. TSN also provides ultra-reliability for critical traffic via frame replication as well as protection against bandwidth violation, malfunctioning, and malicious attacks [47]. In addition, TSN supports frame preemption, which, on the one hand, reduces the latency of critical traffic and, on the other hand, improves the efficiency of bandwidth usage for noncritical messages.

The TSN standards provide a flexible toolbox from which a network designer can pick what is required for designing the targeted application. However, each protocol in this toolbox may not exist independently, and some competing approaches to configuring individual protocols are mutually exclusive and only support individual protocol feature sets. As an overview, here we list some relevant TSN specifications for industrial automation [46], as shown in Figure 3. Their details will be provided in Section 3.

- *IEEE 802.1AS(-Rev)* “Timing and Synchronization for Time-Sensitive Applications” and its revision (IEEE 802.1AS-Rev) are key TSN standards for achieving network-wide time synchronization. IEEE 802.1AS includes several versions that utilize the IEEE 1588 **Precision Time Protocol (PTP)** as the primary profile for synchronization. The amended version, IEEE P802.1AS-Rev, includes enhancements such as support for fault tolerance and scenarios with multiple active synchronization masters.

- *IEEE 802.1Qbv* "Enhancements for Scheduled Traffic", also known as the "**time-aware shaper (TAS)**" [8], introduces the concept of a **time-triggered (TT)** switch. With the help of a centralized scheduler, IEEE 802.1Qbv controls the open or closed status of the gates at the egress of a switch to manage the flow of queued traffic. By following a well-designed schedule, the traffic delay is deterministic at each switch, ensuring that the e2e latency is guaranteed.
- *IEEE 802.1Qav* "Forwarding and Queuing Enhancements for Time-Sensitive Streams", known as the "**credit-based shaper (CBS)**", is designed to limit the transmission bandwidth for multiple streams.[4] By collaborating with the **stream reservation protocol (SRP)**, the CBS shaper can manage the buffer size at the receiving port, providing bounded latency per stream type. Additionally, it can restrict the transmission of audio/video frames to protect best-effort traffic.
- *IEEE 802.1CB* "Frame Replication and Reliability Elimination" provides a mechanism for duplicating streams to enhance reliability, e.g., transmitting a stream over multiple available paths and re-merging the duplicates at the destination. Utilizing IEEE 802.1Qca ("Path Control and Reservation"), the redundancy management in IEEE 802.1CB can set up and manage designated disjoint paths, thereby maintaining full control over the duplicated streams.
- *IEEE 802.1Qcc* "Stream Reservation Protocol Enhancements and Performance Improvements" offers various models for reserving streams on a TSN-enabled network. It supports three resource management models: a fully distributed model, a centralized network/distributed user model, and a fully centralized model. This protocol enables deterministic stream reservation on each intermediate bridge, thereby guaranteeing e2e latency.
- *IEEE 802.1Qbu* "Frame Preemption" (together with IEEE 802.3br) provides a mechanism allowing higher priority frames to interrupt lower priority frames. This ensures that critical traffic is protected from interference by non-critical traffic. Although the TAS shaper in IEEE 802.1 Qbv can mitigate transmission jitter by blocking lower priority queues before the transmission begins, the preemption capability defined in IEEE 802.1 Qbu is essential for further enhancing the real-time performance of critical traffic.

In summary, the IEEE TSN TG focuses on enhancing the reliability and real-time capabilities of the Ethernet standard in industrial automation through a comprehensive set of standards. This includes IEEE 802.1AS for time synchronization, IEEE 802.1Qbv/802.1Qbu/802.3br for traffic shaping and scheduling, IEEE 802.1CB for reliability, and IEEE 802.1Qcc for centralized resource management. In the following, we will delve deeper into how TSN achieves high-precision time synchronization, bounded latency, reliability, and resource management through these standards.

3 TSN Standardization

We can broadly classify the TSN standardization efforts into four major sets, as shown in Figure 3, while the classifications are not disjoint, as some standards contribute to multiple aspects. The four main pillars on which TSN is built are: (1) time synchronization, (2) guaranteed e2e latency, (3) reliability, and (4) resource management. We will detail each aspect below, and explain the advantages of TSN over the existing industrial solutions at the end of this section.

[4] In this paper, we use the terms "flow" and "stream" interchangeably unless otherwise specified.

3.1 Time Synchronization

Time synchronization is crucial for most applications targeted by the IEEE 802.1Q standards. Many TSN standards depend on network-wide precise time synchronization, with varying requirements when transitioning from AVB streaming to time-sensitive and safety-critical control applications. In a typical TSN network, a common time reference is shared by all TSN entities and used to schedule data and control signaling. Time synchronization in TSN is defined primarily by two key standards: IEEE 802.1AS and IEEE 802.1AS-Rev.

The IEEE 802.1AS standard utilizes and optimizes the IEEE 1588-2008 (1588v2) protocol, which includes the **Generic Precision Time Protocol (gPTP)** to synchronize clocks across the network [119]. It is also one of the three IEEE 802.1 AVB standards, targeting network audio/video applications. gPTP achieves clock synchronization between network devices by exchanging predefined messages across the communication medium.

A typical gPTP employs a messaging mechanism between the **Clock Master (CM)**, also known as the **GrandMaster (GM)**, and **Clock Slaves (CS)** to create a time-aware network. This network uses peer-to-peer delay mechanism to calculate timing information such as link latency (between bridges) and residence time (within bridges). Link latency consists of the time spent on the link (e.g., the single-hop propagation delay between two adjacent switches), and residence time includes the time spent within the switch (e.g., processing time, queuing time, and transmission time). The GM clock serves as the reference time at the root of the time-aware network hierarchy and is selected by the **Best Master Clock Algorithm (BMCA)** [157], which automatically designates the grandmaster device. The BMCA dynamically configures the synchronization hierarchy, known as the synchronization spanning tree. This spanning tree is constructed using a priority vector derived from the announce message. Each port is assigned to one of three states: master, slave, or passive. Additionally, ports not in use are set to a disabled state.

In the gPTP protocol, entities are divided into time-aware systems and non-time-aware systems. A time-aware system must implement one or more PTP instances for synchronization across single or multiple domains. A PTP instance is required to support essential functions of the IEEE 802.1AS standard, such as BMCA and synchronization state machine. Depending on its function, a PTP instance is further categorized as either a PTP relay instance, which communicates synchronized time from one PTP port to others, or a PTP end instance, which has only one PTP port.

IEEE 802.1AS-Rev introduces new capabilities required for time-sensitive applications in several ways. First, GMs and synchronization trees can be redundantly configured to enhance fault tolerance, allowing synchronization trees to be explicitly configured without using the BMCA algorithm. Additionally, IEEE 802.1AS-Rev supports redundant communication by enabling multiple time domains for gPTP. Each gPTP domain operates as a separate instance, allowing network devices to execute multiple instances of gPTP simultaneously. This enhances redundancy by permitting multiple grandmaster clocks and synchronization spanning trees, facilitating seamless synchronization recovery.

3.2 Bounded Latency

One primary characteristic of TSN standards is the guaranteed delivery of messages with stringent timing constraints, i.e., bounded e2e latency. In this section, we discuss several standards in TSN towards bounded latency.

3.2.1 IEEE 802.1Qav Forwarding and Queuing of Time-Sensitive Streams. IEEE 802.1Qav specifies the enhancements for the transmission selection algorithms of Ethernet switches and defines the **credit-based shaper (CBS)** to ensure bounded latency for time-sensitive traffic by regulating the transmission rate. CBS is a traffic shaping mechanism that regulates bandwidth allocation

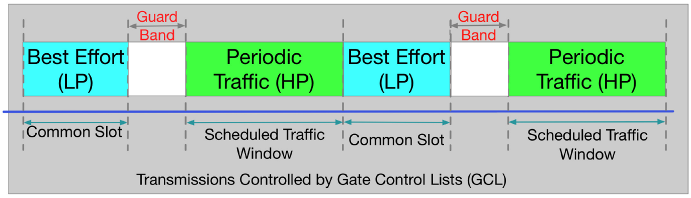

Fig. 4. Overview of IEEE 802.1Qbv Time-Aware Shaper (TAS): the scheduled traffic will be sent over TDMA-like synchronized slots. HP traffic has guaranteed reserved resources across the network, while LP traffic is best-effort low-priority traffic.

for high-priority-shaped queues to reduce delays in medium- and low-priority unshaped queues, thereby enhancing fairness. In CBS, each output queue is associated with a credit counter. The credit counter accumulates credits when the queue waits to transmit frames and consumes credits when frames are transmitted. A frame can only be transmitted if the credit of its queue is non-negative and no other frames are being transmitted at the same time. If no frames are waiting for transmission, the credit of the queue is reset to zero. The queue credit decreases and increases at a constant rate which is configurable.

For bandwidth-intensive applications, the CBS protocol can establish an upper bound for each traffic class, ensuring that no traffic class exceeds the pre-configured threshold on reserved bandwidth, typically less than 75% of the maximum bandwidth. Along with SRP, the CBS shaper aims to limit delays to less than 250 μs per bridge and the worst-case latency to up to 2 *ms* for class A, and up to 50 *ms* for class B in a simple network setup [73]. However, these delay scales may still be too high for industrial applications. This has motivated the TSN TG to introduce other standards, such as IEEE 802.1Qbv, IEEE 802.1Qch, and IEEE 802.1Qcr, to meet the stringent timing requirements of industrial applications.

*3.2.2 IEEE 802.1Qbv Enhancements to Traffic Scheduling (**Time-Aware Shaper (TAS)**).* IEEE 802.1Qbv introduces the concept of a gate per queue to control the open/close of a queue, where a frame can be transmitted only if the gate of the corresponding queue is open. In TAS, critical traffic is scheduled in protected traffic windows with allocated time slots, similar to the TDMA paradigm. Each window can have an allotted transmission time for high-priority traffic, as illustrated in Figure 4. To prevent potential interference, the traffic windows are isolated by a specified time duration, called the guard band. The guard bands enforce time intervals after best-effort traffic during which all gates are closed, ensuring neither best-effort traffic nor periodic traffic can be sent during these intervals. These guard bands are required to prevent large best-effort frames from interfering with periodic traffic.

The TAS shaper requires that all traffic windows be well synchronized and scheduled among all the time-aware bridges. The communication schedule in IEEE 802.1Qbv is realized by the scheduled gate mechanism, which controls the opening and closing of queues using a pre-determined **gate control list (GCL)**. Each GCL includes a limited number of entries, with each entry providing the status of associated queues over a particular duration. The GCL repeats itself periodically, and this period is called the cycle time. The network-wide schedule is generated by **centralized network configuration (CNC)** and deployed on individual bridges. Although the IEEE 802.1Qbv standard defines the scheduling mechanism of TAS, its configuration, i.e., what to put in the GCL and how to assign queues for individual traffic at each hop, lacks a clear-cut best practice [155]. This has resulted in significant efforts from both researchers and practitioners to study the TAS-based scheduling problems in various industrial applications. More discussion regarding TAS scheduling is provided in Section 4.3.

3.2.3 IEEE 802.3br and 802.1Qbu Interspersing Express Traffic and Frame Preemption. To address the inverted priority problem, i.e., ongoing transmission of a low-priority frame prevents the transmission of high-priority frames, the IEEE 802.1 TG along with the IEEE 802.3 TG defined the frame preemption protocol in IEEE 802.1Qbu and IEEE 802.3br. These technologies work together to effectively manage traffic using changes to both the MAC scheme, as controlled by IEEE 802.3, and management mechanisms, as supervised by IEEE 802.1. The frame preemption capability can be combined with any traffic management algorithms defined in IEEE 802.1Q, such as the TAS shaper and CBS shaper, to enhance determinism and real-time performance for critical traffic.

IEEE 802.1Qbu allows time-critical data frames to be split into smaller fragments and preempt the non-critical frames on the same physical link, even if they are in transition. This frame preemption scheme divides an egress port into two distinct interfaces based on the MAC layer: **preemptable MAC (pMAC)** and **express MAC (eMAC)** [96]. The pMAC targets preemptable frames, while the eMAC targets preemptive frames. An incoming frame is mapped to only one egress interface according to the frame preemption status table, with the default option being the eMAC.

IEEE 802.3br introduces an optional sublayer called the MAC Merge sublayer, which attaches an eMAC and a pMAC to the PHY layer through a reconciliation sublayer [174]. The PHY layer remains unaware of the preemption, while the MAC Merge sublayer and its MACs support frame preemption as defined in IEEE 802.1Qbu. The MAC Merge sublayer provides two approaches to manage the transmission of preemptable traffic alongside express traffic. One approach interrupts (preempts) the preemptable traffic currently being transmitted, while the other prevents preemptable traffic from being transmitted in the first place.

*3.2.4 IEEE 802.1Qch **Cyclic Queuing and Forwarding (CQF)**.* The IEEE 802.1Qch standard introduces the CQF mechanism, also known as the **Peristaltic Shaper (PS)** [127]. CQF is an efficient forwarding scheme proposed to simplify the design of a TSN switch, and it can deliver predictable and deterministic e2e latency [101]. It is designed for limited-scale networks with time synchronization. Among the eight queues of a port of each switch, CQF reserves at least two queues performing enqueue and dequeue operations in a cyclic manner. Figure 5 shows an example of CQF operation on a chain topology with two switches SW1 and SW2. Time is divided into equal cycles with the length of T, which is delimited by the red vertical lines. During the first interval (i.e., cycle x), frames A, B, and C are sent out by end station ES1 and arrive at SW1, enqueuing them in q_1. In the following interval (i.e., cycle $x + 1$), these frames are dequeued and forwarded to SW2, stored in q_1. Meanwhile, another two frames D and E arrive at SW1, enqueued in another queue q_2. The operation repeats in each cycle. CQF can provide a deterministic e2e latency guarantee since it follows two principles. (1) The sending cycle of a frame on a switch and the receiving cycle on the subsequent switch are the same. (2) Any frame received by a switch on cycle x must be sent out on the next cycle $x + 1$. Thus, the e2e latency of a frame is determined by the routing path length and cycle size T.

The frame preemption scheme can also work together with CQF to shorten the cycle time of frame transmission, as the size of a frame fragment is smaller than that of a full frame. To make CQF work properly, all frame fragments must be received within the scheduled time cycle. Accordingly, to guarantee bounded and deterministic latency, it is crucial to carefully design the cycle length along the routing path. Due to its simplicity, CQF can be easily supported by extending a standard Ethernet switch with statically configured queues.

*3.2.5 IEEE 802.1Qcr **Asynchronous Traffic Shaping (ATS)**.* The TAS shaper can provide deterministic real-time communication in a TSN network but requires high-precision network-wide time synchronization. However, industrial networks may suffer from timing misalignment, such as drift or skew in timing signal frames, lost timing frames, and inaccuracy, which can

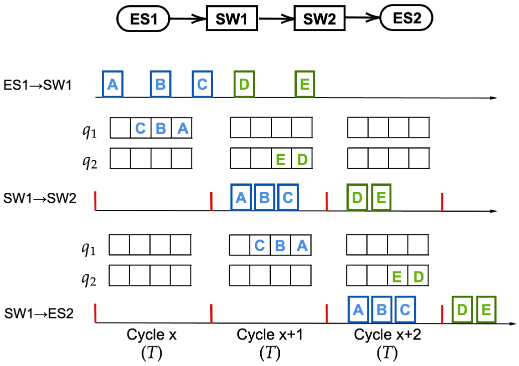

Fig. 5. An example of CQF operation on a chain topology with two switches. Time is divided into cycles with the length of T. Frames received by a switch in cycle x will be sent out in the next cycle $x + 1$.

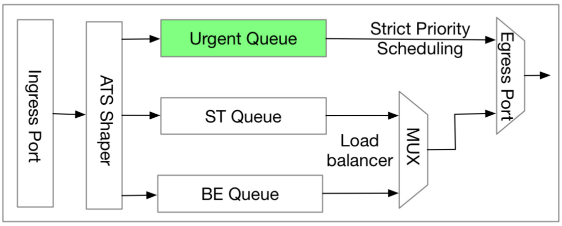

Fig. 6. An ATS bridge example: ATS shaper determines the type of traffic at the ingress. ST represents the high-priority scheduled traffic, and BE represents the low-priority best-effort traffic [85].

cause asynchrony. This issue worsens with the increasing scale of the network [172]. To address this, IEEE 802.1Qcr aims to smooth out traffic patterns by reshaping TSN streams per hop and prioritizing urgent traffic over non-deterministic traffic. The ATS shaper works asynchronously, not requiring synchronization on traffic transmission, and relies heavily on an **Urgency Based Scheduler (UBS)**. The UBS prioritizes urgent traffic by queuing and reshaping each individual frame at each hop. Asynchronicity is achieved through a **Token Bucket Emulation (TBE)** and an interleaved shaping algorithm to eliminate burstiness. The TBE controls traffic by the average transmission rate but allows a small portion of burst traffic to occur. Figure 6 shows an example of an ATS shaper. The ATS shaper determines the traffic types at the ingress port for each incoming traffic. In the case of urgent traffic, it will be assigned to an urgent queue, which follows strict priority scheduling. For traditional high-priority scheduled traffic and low-priority best-effort queues, they follow a fair multiplexed transmission scheme.

Table 1 provides a summary of different TSN shapers. In the table, ‘Synchronization’ represents the network model, which can be either synchronous or asynchronous, and ‘/’ indicates that it does not require time synchronization. ‘Main Tech’ refers to the main technology the shaper uses, e.g., TDMA. ‘Topology Dependence’ indicates whether the e2e latency is influenced by the adopted network topology. ‘Trigger’ represents the triggering mechanisms of the shaper.

3.3 Reliability

Ultra-high reliability is another fundamental QoS requirement for industrial critical traffic. To achieve this, TSN provides several mechanisms to exploit the spatial redundancy of the communication channel and transmit replicated frames through multiple channels to tolerate both permanent and temporary faults. For this purpose, several standards have been defined in TSN,

Table 1. Summary of Different Shapers

Shaper	Full name	Synchronization	Main Tech	Topology dependence	Trigger
TAS (Qbv)	Time-Aware Shaper	Sync.	TDMA	Dependent	Cycle
CBS (Qav)	Credit-based Shaper	/	Credit-based Shaping	Dependent	Credit
PS (Qch)	Peristaltic Shaper	Sync.	Double Buffering	Independent	Cycle
ATS (Qcr)	Asynchronous Traffic Shaper	Async.	Event-Trigger	Dependent	Event

including IEEE 802.1CB and IEEE 802.1Qca. The IEEE 802.1CB standard manages creating and eliminating frame replicas to be transmitted through the existing path(s), while IEEE 802.1Qca allows for creating and managing multiple paths between any pair of nodes in the network. Besides, the IEEE 802.1Qci standard defines frame filtering and policing operations.

3.3.1 IEEE 802.1CB ***Frame Replication and Elimination for Reliability (FRER)***. The IEEE 802.1CB standard lowers packet loss probability by replicating transmitted packets, sending them on disjoint network paths, and reassembling replicas at the receiver. IEEE 802.1CB is a self-contained standard that guarantees reliable and robust communication among applications through proactive measures to tolerate frame losses. Specifically, IEEE 802.1CB includes features such as sequence numbering, replication of each packet in the source station and/or network relay components, transmission of duplicates across separate paths, and elimination of duplicates at the destination and/or other relay components. By sending duplicate copies of critical traffic across disjoint network paths, IEEE 802.1CB minimizes the impact of congestion and failures, such as cable breakdowns. The duplicates are eliminated based on the sequence numbers carried in the frames. To enhance robustness and cope with errors, such as those caused by a stuck transmitter repeatedly sending the same packet, a recovery function is defined to remove packets with repeated sequence numbers.

3.3.2 IEEE 802.1Qca ***Path Control and Reservation (PCR)***. The IEEE 802.1Qca standard builds on two schemes: the **Type-Length-Value (TLV)** extension and the **IS-IS (Intermediate System to Intermediate System)** protocol. The TLV extension is based on the **Link State Protocol (LSP)** of IETF, while the IS-IS protocol is used to establish connections among stations along the transmission path. This enables the IS-IS protocol to control bridged networks, extending the capabilities of the **shortest path bridging (SPB)** to manage multiple routes on the network [133]. IEEE 802.1Qca provides mechanisms for bandwidth allocation and improves redundancy through various methods, such as protection schemes based on multiple redundant trees, local protection for unicast data flows based on loop-free alternates, and restoration after topology changes (e.g., following a failure event).

3.3.3 IEEE 802.1Qci ***Per-Stream Filtering and Policing (PSFT)***. The IEEE 802.1Qci standard defines protocols and procedures for filtering, policing, and service class selection on a per-stream basis. Filtering and policing functions include stream filters, stream gates, and flow meters to determine whether each frame is allowed to pass through to the egress queue. By setting up filtering rules and monitoring the passing frames, the standard can perform mitigation actions if violations are detected. Thus, IEEE 802.1Qci provides QoS protection when multiple streams share the same egress queue of a switch, preventing interference among them [18]. In addition, it improves network security against DoS attacks by identifying and dropping unauthorized or malicious transmissions, enhancing network robustness.

3.4 Resource Management

Resource management is another key aspect of TSN to ensure the efficient allocation and utilization of network resources to meet the stringent requirements of industrial applications. It involves various mechanisms and protocols to manage network bandwidth, prioritize traffic, and maintain

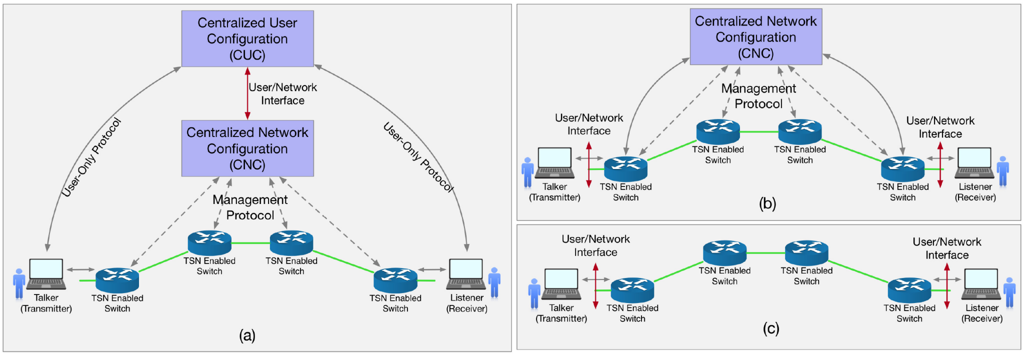

Fig. 7. Three TSN resource management models: (a) fully centralized model; (b) centralized network / distributed user model; (c) fully distributed model.

QoS through the definition of several standards, including IEEE 802.1Qcp, IEEE 802.1Qcc, and IEEE 802.1CS.

3.4.1 IEEE 802.1Qcp YANG Data Model. IEEE 802.1Qcp defines a **YANG (Yet Another Next Generation)** data model, specifying a data modeling language used to model configuration data and state data manipulated by network management protocols such as NETCONF and RESTCONF. Using the YANG model, IEEE 802.1Qcp allows configuration and status reporting based on **Unified Modeling Language (UML)** to manage IEEE 802.1 bridge devices. YANG models the hierarchical organization of data as a tree, with each node representing configuration data, state data, **RPC (remote procedure call)** operations, and notifications. A set of related data nodes are organized into a module, the primary building block of the YANG model [22]. To simplify the maintenance and management of complex modules, each module can be further subdivided into submodules. The industry-wide implementation of the YANG model provides a universal interface to integrate resource management across diverse devices and equipment to fulfill the TSN standards.

3.4.2 IEEE 802.1Qcc SRP Enhancements and Performance Improvements. The IEEE 802.1Qcc standard is an enhancement of the **Stream Reservation Protocol (SRP)** (IEEE 802.1Qat) and deals with the configuration of TSN networks. IEEE 802.1Qat, originally designed for CBS shaper, manages the registration and reservation of resources within each bridge (e.g., buffers and queues) along the traffic path between the talker and the listener. Specifically, it serves as an admission control protocol where the talker registers the sending traffic with the required bandwidth, and it will be granted permission or not, depending on resource availability. This enables QoS management for streams with specific latency and bandwidth requirements.

IEEE 802.1Qcc amends the IEEE 802.1Qat standard by extending the capabilities of SRP to adopt more complex shaping mechanisms, such as TAS with frame preemption. IEEE 802.1Qcc defines a **user-network interface (UNI)**, which provides an abstract functionality between end stations (i.e., user side) and bridges (i.e., network side). The high-level idea is that the user specifies the requirement for the streams they want to transmit without knowing all the details about the network, and the network analyzes this requirement along with network capabilities and configures the bridges to meet the user requirements. IEEE 802.1Qcc defines three configuration models [129], as shown in Figure 7: the fully centralized model, the centralized network/distributed user model, and the fully distributed model. The fully centralized model introduces **Centralized User Configuration (CUC)** as the centralized manager for end users and provides the user requirements to the CNC through UNI. In the centralized network/distributed user model, the CNC configures

TSN elements according to user requirements provided by the end bridges connecting end stations through UNI. In the fully distributed model, there is no centralized network configuration entity, and the network is configured in a fully distributed manner.

3.4.3 IEEE 802.1CS ***Link-Local Reservation Protocol (LRP)****.* The IEEE 802.1CS standard facilitates the replication of a registration database within a network link, i.e., from the device at one end to the device at the other end of the link. This enhances communication regarding resource registration among point-to-point devices and enables dynamic discovery, registration, and management of resources at a local level. The current 802.1Q Multiple Registration Protocol supports databases up to 1500 bytes and significantly slows down when handling larger databases. To address this limitation, LRP is optimized to support the replication of registration databases on the order of 1 Mbyte. This enhancement enables new applications requiring much larger data sizes for configuration, registration, and reservation. LRP improves resource management efficiency since it operates within the local network segment without centralized management.

3.4.4 IEEE 802.1Qdd ***Resource Allocation Protocol (RAP)****.* IEEE 802.1Qdd defines RAP, which uses LRP from IEEE 802.1CS to support dynamic resource reservation for unicast and multicast streams in the fully distributed model. RAP also provides support for accurate latency calculation and reporting, and it is not limited to bridged networks. It aims to address issues present in the current IEEE 802.1Q **Multiple Stream Reservation Protocol (MSRP)**, which has limitations in terms of the number of reservations, admissions, and configuration size in distributed stream reservation scenarios [92]. As of this writing, the standardization of RAP is still ongoing (IEEE P802.1Qdd Draft 0.9).

The advantages of TSN compared to existing industrial solutions. After detailing the major capabilities of TSN, here we summarize its advantages over the existing Ethernet-based fieldbus systems. These advantages include openness, interoperability, convergence, and performance. First of all, openness and standardization are crucial to industrial automation since they promote wide cooperation among industrial partners. TSN is an open and standardized IEEE technology that is unaffiliated to any organization or company, and thus, the major manufacturers are very active in promoting TSN. Second, TSN ensures vendor-independent interoperability among the industrial devices, avoiding vendor lock-in and enabling system-wide connectivity. The combination of OPC UA and TSN, described in the following section, further fulfills the communication all the way from the sensor to the cloud. Moreover, TSN enables the convergence of IT and OT, which were previously kept separate in traditional industrial Ethernet-based protocols. Breaking down the communication barriers between IT and OT makes accessing data from industrial subsystems easier, where different traffic types can coexist in the network with their specific QoS requirements being met. In addition to the above advantages, TSN also excels in performance. While some advanced Ethernet-based protocols, e.g., PROFINET IRT, can also achieve deterministic real-time performance, TSN surpasses these solutions in latency (cycle time below 50 microseconds), jitter (less than ±100 nanoseconds), and scalability (more than 10,000 network nodes) [26]. Therefore, its openness, vendor-neutral interoperability, IT/OT integration support, and higher network performance, make TSN a highly effective and reliable choice for modern industrial automation.

4 Integrating TSN into Industrial Automation

In this section, we first detail the key benefits of TSN for industrial automation and highlight the opportunities for integrating TSN into industrial automation through potential system-level integration. We then elaborate on TSN traffic scheduling for achieving deterministic timing guarantees. At last, as a crucial step before deploying TSN in real fields, we discuss the importance of TSN testbeds, highlighting their role in validating TSN performance in real-world industrial environments.

4.1 Why Do We Need TSN in Industrial Automation?

TSN is a game-changing technological advancement based on Ethernet, and it is set to reshape the industrial communication landscape. This is mainly due to the many benefits offered by TSN to modern industrial automation networks, e.g., interoperability, convergence, and determinism.

As described in Section 2.1.1, the connectivity of industrial devices, i.e., interoperability, plays a critical role in industrial automation. At present, there are many tailored protocols and customized devices on the market for industrial Ethernet-based applications. While in many industrial application scenarios, customers may select different industrial Ethernet protocols to deploy their devices. This results in protocol incompatibility and leads to vendor lock-in, which leaves the customers with only two options. One is to purchase all their devices from the same vendor even though some are not their best choices. The other option is to purchase their devices from multiple vendors but develop a convertible solution to integrate the devices, e.g., by implementing gateways to adapt among various industrial Ethernet protocols. However, both options are costly and can limit innovation on the factory floor [25]. Given the strength of TSN as an open IEEE standard, it guarantees compatibility at the network level among devices from different vendors. With TSN, a network consisting of multiple-vendor devices can inter-operate and be configured via a single standard interface. This provides customers with more options to build their system, avoids vendor lock-in, and enables connectivity across systems. The standardized network structure also leads to a lower cost of ownership since the customers only need to replace existing switches with TSN switches instead of duplicating networks and maintaining the additional hardware and software.

The IT/OT integration, accelerated by the rapid development of advanced manufacturing, acts as another critical enabler in the automation industry [107]. In legacy industrial Ethernet-based networks, different communication needs for IT and OT hinder the integration of these two fields. Specifically, a larger bandwidth is typically required for data communication in the IT fields, while deterministic performance is the key for OT involving control operations. On the other hand, the digitization trend of industrial automation requires all types of data information (e.g., analog signals, sounds, images, and texts) must be converged. To this end, TSN provides the capability to break down communication barriers between various subsystems, including critical and non-critical systems. Different traffic types can coexist and be transmitted over the same network with no impact on traffic with a higher criticality level from traffic with lower priority. Network convergence provided by TSN makes it easier to access data from industrial systems and send them to the enterprise systems over standard Ethernet or the other way around without the need for gateways.

Despite handling various traffic types across numerous devices in such converged networks, TSN can still provide deterministic performance guarantees, especially for critical traffic. TSN ensures that the timing of critical traffic is predictable and consistent, which is essential for industrial automation applications. With deterministic message delivery, devices can communicate in real time, simplifying the configuration of systems, devices, and applications and increasing productivity by enabling the machines to run cooperatively rather than independently. Informed decision-making by humans or other machines can also be processed in real time. This benefit of deterministic communication is achieved through TSN traffic scheduling based on network-wide time synchronization, which will be elaborated in Section 4.3.

4.2 TSN-based Converged Industrial Networks

TSN standardizes a set of technologies within the framework of IEEE 802.1 to provide guaranteed QoS. It is worth noting that TSN only resides at Layer 2 of the OSI model, i.e., it aims to provide bounded latency and jitter for point-to-point communication. Thus, TSN is not a complete communication protocol but rather can be taken as a building block to provide the determinism foundation for converged industrial networks and it needs to be used in combination with

Table 2. Summary of Different TSN-based Converged Industrial Networks

Organization or authors	Type	Year	Technology	Summary
PROFIBUS & PROFINET International (PI) [99]	White paper	2021	PROFINET	Principles, use cases, and architecture
Schriegel et al. [108]	Research paper	2021	PROFINET	Ethernet bridging mode
Karl Weber [145]	White paper	2018	EtherCAT	Integration approach
Balakrishna et al. [16]	Research paper	2021	EtherCAT	Simulation-based case study
Woods et al. (ODVA) [149]	Research paper	2017	EtherNet/IP	Use cases and challenges
Hantel et al. (ODVA) [60]	Research paper	2022	EtherNet/IP	Technical recommendations
CC-Link Partner Association (CLPA) [32]	White paper	2023	CC-Link	Technical specification
Li et al. [72]	Research paper	2020	OPC UA	Architecture and implementation
Pfrommer et al. [98]	Research paper	2018	OPC UA	Messaging mechanism and implementation
Gogolev et al. [55]	Research paper	2018	OPC UA	Field device case study

higher-layer protocols to provide end-to-end QoS guarantee. On the other hand, industrial automation requires the Ethernet to support the convergence of all kinds of networks and traffic types typically found in an industrial setting.

Converged networks in industrial settings require flexibility and scalability to use the same infrastructure (including small devices like sensor nodes, machine, and production line control devices, as well as big devices like data servers) for concurrent transmission of deterministic real-time communication (e.g., OT traffic) and non-deterministic best-effort communication (e.g., IT traffic). TSN is deemed as a key enabling technology to establish converged industrial networks with the following two trends [138]: (1) Fieldbus[5] over TSN, and (2) OPC UA over TSN. Table 2 gives a summary of representative TSN-based converged industrial network solutions. Their details are described below.

4.2.1 Fieldbus over TSN. At present, the industrial communication market is still dominated by Ethernet-based fieldbus systems, and there are many different fieldbus solutions in the market, e.g., PROFINET, EtherNet/IP, EtherCAT, Powerlink, and CC-Link. A major obstacle for today's Ethernet-based fieldbus systems is that they do not fulfill the convergence requirement of emerging industrial automation applications (e.g., a close IT/OT integration). Thus, combining industrial fieldbuses with TSN provides a way that can accomplish such requirements. There exist two main approaches for transmitting industrial fieldbus communication over TSN. One approach is to set up a new TSN network in accordance with every specification of the newly defined IEEE standards over Layer 1 and Layer 2 of OSI in factory networks so that fieldbuses can be transmitted without alternation. The other approach is to install active network gateways to convert all other network traffic between them to TSN-compatible Ethernet frames [138].

Many fieldbus providers are already offering their products mapped to TSN, enabling seamless integration. For example, PROFINET over TSN [99] makes use of TSN features and supplements PROFINET on the Ethernet layer with IEEE standardized counterparts. With TSN, PROFINET is standing on a robust and future-proven foundation, which in turn creates more planning reliability for production and industrial solutions. On the other hand, existing PROFINET services (e.g., diagnostics and parameterization) and profiles (e.g., PROFIsafe, PROFIenergy, PROFIdrive) work as before on top of PROFINET over TSN and do not require any changes from the user.

EtherCAT over TSN [145] defines a seamless adaptation to use both technologies and capitalize on their respective advantages without requiring any changes to the EtherCAT slaves. Adding EtherCAT segments as structuring elements in TSN reduces the complexity in backbones by using shared frames for a group of slaves and enabling internal configuration for a machine. TSN will protect EtherCAT segments from unwanted traffic while increasing the efficiency of the combined

[5] Here we refer to Ethernet-based fieldbus systems.

EtherCAT-TSN system. Combined EtherCAT and TSN can enhance flexibility at the automation cell level while maintaining total control of the various automation tasks.

ODVA, which is a standards development organization and membership association, presents a recommended high-level approach for incorporating TSN capability into EtherNet/IP and identifies several major technical aspects of EtherNet/IP over TSN [60]. TSN will be introduced in ODVA technologies as an optional and backward-compatible Data Link Layer for the EtherNet/IP implementation of **CIP (Common Industrial Protocol)**.

CC-Link IE TSN [32] is an open industrial network utilizing TSN to seamlessly connect information systems to production sites. With TSN, CC-Link IE TSN is able to increase openness while further strengthening performance and functionality. In addition to the above solutions with individual fieldbus systems, [34] designs a hybrid wired/wireless protocol conversion module that can realize intercommunication of three industrial Ethernet such as PROFINET, EtherCAT, and Ethernet/IP, and proposes a TSN-compatible frame to communicate with TSN based gateway.

4.2.2 OPC UA over TSN. Today's proprietary Ethernet-based fieldbus systems are broadly applied across different industrial automation networks to meet specific topology requirements, communication speeds, or latency guarantees. However, these communication protocols are often incompatible, resulting in fragmented networks that cannot seamlessly communicate with each other. OPC UA [70] was developed to solve this problem by allowing industrial devices operating with different protocols and on different platforms (e.g., Windows, Mac, or Linux) to communicate with each other. OPC UA supports two communication models, client-server (point-to-point communication based on TCP/IP) and publisher-subscribers (one-to-many communication supported by the new PubSub extension), without real-time capability. Thus, in conjunction with TSN, OPC UA over TSN under the pub/sub communication model allows deterministic transmission of real-time data and offers the flexibility and openness inherent to OPC UA [131]. Note that, OPC UA over TSN and the above discussed fieldbus over TSN systems clearly overlap, but they are not replacing each other but will likely coexist for a long while. This is mainly due to the following fact. The strength of OPC UA, with real-time communication enabled by TSN, is that it allows different networks to communicate, especially at the factory- and enterprise-level. Industrial Ethernet, on the other hand, is primarily designed for communication between field devices and controllers. Below, we briefly discuss some OPC UA over TSN solutions.

[72] proposes a communication architecture using the OPC UA and TSN for manufacturing systems. The proposed OPC UA TSN is a two-tier communication architecture, including the upper factory-edge tier and the lower edge-field tier. TSN is adopted as the communication backbone to connect different control subsystems in the field layer and the entities of the upper layers. OPC UA is adopted to realize horizontal and vertical information exchange between the entities of each layer. [98] presents an OPC UA PubSub over TSN, which enables TSN to be used for the transport of OPC UA PubSub messages in practice. In the proposed approach, the message for the publisher is prepared in a (hardware-triggered) interrupt to ensure short delays and small jitter. Specific modifications are performed to allow the interaction between a best-effort standard OPC UA server and a real-time OPC UA PubSub publisher with access to a shared information model. The approach was implemented in open source based on the open62541 OPC UA SDK. [55] presents a case study on a TSN-enabled OPC UA integration for a field device. The evaluation indicates that the OPC UA integration of the field devices can be implemented using COTS software and hardware components. These R&D efforts validate the potential of OPC UA TSN as a vendor-independent successor technology. OPC UA TSN is expected to quickly reveal itself as a game changer in the field of industrial automation, becoming the promising candidate to establish a holistic communication infrastructure from the sensor to the cloud [26].

4.3 Traffic Scheduling

As described in Section 3, the TSN TG has developed a suite of traffic shapers in the TSN standards, including TAS, CBS, PS, and ATS (see the summary in Table 1). These shapers provide a toolkit for managing network traffic to meet the diverse timing requirements. Among these shapers, TAS stands out and draws special attention due to its ability to achieve deterministic timing guarantees by leveraging network-wide synchronization and time-triggered traffic scheduling mechanisms [171], making it a key enabler to support deterministic real-time traffic in industrial automation.

A TSN switch is equipped with a set of time-gated queues to buffer frames from different traffic flows, and the control of the queues is specified by a predefined GCL. In addition, the priority filter in each switch utilizes a 3-bit **Priority Code Point (PCP)** field in the packet header to identify the stream priority and directs incoming traffic to the specific egress queue according to the priority-to-queue mapping. The configuration of GCL and traffic-to-queue mapping together define the network-wide schedule, which is determined by CNC and deployed on individual switches to guarantee the timing requirements of all time-triggered traffic. Traffic scheduling is thus one of the most critical problems in TSN, resulting in a large amount of research effort to develop various novel scheduling methods.

Industrial applications that employ TSN as the communication fabric can be diverse regarding traffic patterns, network topology, deployment environment, and QoS requirements. Consequently, the specific TSN scheduling problem to be studied may vary significantly from the perspectives of the network model, traffic model, and scheduling model.

— The *network model* defines key attributes of the directed logical links in TSN, such as the propagation delay on Ethernet cables, processing delay on switches, link rate, number of available queues, and maximum GCL length. These parameters are typically determined by the capacity of the TSN switch or end station connected to each link.

— The *traffic model* defines the parameters characterizing each TSN flow, including release time, period, payload size, deadline, and jitter. Each parameter can be individually modeled to capture the targeted traffic type based on specific industrial application scenarios. For example, the traffic model can be classified into fully scheduled or partially schedulable traffic, depending on whether the release time of flows is predefined or determined by the corresponding talker. Additionally, based on jitter requirements, the traffic model can be categorized as a zero-jitter model or a jitter-allowed model.

— The *scheduling model* specifies the constraints on the TSN system, including queuing delay, scheduling entity, routing and scheduling co-design, fragmentation, and preemption. For instance, based on assumptions regarding queuing delay, scheduling models can be classified into no-wait and wait-allowed models. The scheduling entity determines whether the model is frame-based or window-based. Furthermore, depending on whether the routing path of each traffic flow is predefined or needs to be determined, scheduling models can be categorized as fixed routing models and joint routing and scheduling models.

Based on the above TSN model categorization, in a most recent TSN survey [155], we present a systematic review and experimental study on 17 representative TAS-based TSN scheduling methods comparing their performance using various metrics.[6] This work offers comprehensive experimental comparisons among selected scheduling methods, including a diverse set of TSN system models and algorithms focusing on real-time scheduling of time-triggered traffic. The comparison

[6] The established benchmark for performance evaluation of TSN scheduling methods is open-sourced. Please refer to our technical report [154] and GitHub repository [152]. We encourage the community to utilize this open-source toolkit to evaluate their scheduling methods to boost the development of TSN-related R&D projects.

results demonstrate that there is no one-size-fits-all scheduling method that can achieve dominating performance in all scenarios. Furthermore, diverse experimental settings complicate the fair evaluation of scheduling methods without introducing bias, which can make conclusions from previous studies only valid under specific settings. These findings also validate the inherent complexity of TSN traffic scheduling which is still an open problem.

4.4 TSN Testbeds

With all the benefits of TSN for industrial automation, before its deployment in real-world industrial sites, a crucial step is to validate its performance on ensuring all the stringent requirements posed by industrial automation applications. In general, three primary methods are used for evaluating TSN protocols and systems: theoretical analysis, simulation, and hardware testbeds [132]. Many theoretical analysis frameworks have been developed to evaluate TSN, e.g., [58, 79, 159]. However, these analysis frameworks make certain assumptions and abstract the behaviors of TSN systems compared to real-world settings. Simulation-based evaluation is another popular option, and simulation tools, e.g., OMNeT++ and NS-3, have been widely used in TSN research [38, 45, 93]. The advantages of simulations include flexibility, reduced cost, and scalability. However, they do not involve real hardware components, making it impossible to showcase the applicability in real industrial settings. Thus, a high-fidelity way is to use a dedicated physical testbed based on real hardware to conduct well-defined experiments.

Physical testbeds offer many benefits to the design and evaluation of TSN systems, enabling researchers and developers to explore, validate, and optimize their TSN solutions. The solutions can be rigorously evaluated in a controlled environment, ensuring that they meet the stringent industrial requirements. TSN testbeds also facilitate the assessment of interoperability between devices from different vendors. In addition, they help identify and address network configuration challenges and cybersecurity vulnerabilities, thereby mitigating deployment risks and ensuring a smooth transition to TSN-enabled industrial networks. However, the development of a TSN testbed is challenging from different points of view, ranging from implementation costs, sharing capability, and fidelity. Moreover, replicating real-world industrial conditions in a controlled testbed environment is difficult, and the cost and resource requirements, including specialized hardware, software, and skilled personnel, can be significant.

Since TSN is a family of standards, TSN-related testbeds can be built to study different TSN aspects, including traffic scheduling, packet processing, communication over-the-air, performance measurement, and network configuration. There have been a number of TSN testbeds developed for industrial applications and they can be generally classified into (1) general TSN testbeds, (2) OPC UA TSN testbeds, and (3) wireless TSN testbeds. General TSN testbeds (e.g., [40, 102, 132]) focus on the fundamental TSN functions, e.g., scheduled traffic, credit based shaper, and time synchronization, to achieve real-time communication and deterministic behavior. OPC UA TSN testbeds (e.g., [26, 109]) evaluate the integration of OPC UA and TSN to ensure the seamless flow of information among devices from multiple vendors. Wireless TSN testbeds (e.g., [65, 123]) are built to explore the possibility of extending TSN capabilities to wireless media, including Wi-Fi and 5G. We will discuss the opportunities of wireless TSN in Section 6.3, and readers can refer to [169] for more details on the current TSN-related testbeds.

5 Challenges

This section summarizes a number of challenges inherent to TSN standards that should be addressed. We follow the structure of Section 3 to discuss the specific challenges associated with each of the four pillars, i.e., time synchronization, latency guarantee, reliability, and resource management.

5.1 Time Synchronization

Network-wide time synchronization is the foundation of all TSN features aimed at achieving deterministic real-time communication. IEEE 802.1AS is defined within TSN to provide accurate time synchronization using the gPTP protocol as described in Section 3.1. In the following, we discuss several key challenges that impact the accuracy and reliability of time synchronization, e.g., fault tolerance, synchronization overhead, and multi-level hierarchy.

One of the primary challenges in TSN is to maintain precise synchronization across all network devices when applying the master-slave-based gPTP protocol. In a multi-hop TSN network, synchronization errors can occur, leading to synchronization failures [97]. These errors include time value error, i.e., incorrect time-related information (e.g., timestamp error) carried in propagated messages between nodes, and asymmetry in network delay, where the time difference between transmission delays from master to slave and vice versa causes errors [139]. Clock drifts, due to the frequency drift of crystal oscillators, can cause gradual deviation of time clocks in various nodes over time, resulting in synchronization errors. In addition, security attacks, where compromised devices in the synchronization spanning tree propagate erroneous time information, can also lead to accumulated errors and synchronization failures.

To enhance resilience to synchronization failures, IEEE802.1AS only provides a basic level of redundancy, relying on **BMCA (Best Master Clock Algorithm)** to switch to a new **Grandmaster (GM)**. To address this problem, IEEE P802.1ASdm [1] defines a hot standby mechanism to maintain two time domains simultaneously without relying on BMCA [156]. While, addressing synchronization failures may require additional frequent message exchanges on timing information, consuming communication bandwidth and potentially causing back pressure on the centralized control plane, especially in large-scale applications [86]. A trade-off between the synchronization accuracy and incurred overhead should be investigated where the settings of sync messages (e.g., transmission period) can be optimized.

Moreover, industrial automation networks introduce further complexity with multi-level hierarchies on network switches, where different hierarchies may have varied synchronization quality. Since TSN standards operate at the MAC layer, even slight time slips in the upper layer can significantly affect the lower layer. The heterogeneity and accuracy differences among connected devices make a fully centralized time synchronization solution difficult to achieve in large-scale industrial automation. Therefore, applying a time synchronization scheme in industrial automation requires consideration of both network hierarchy and topology, which impacts the propagation mechanism of the synchronization messages.

5.2 Latency Guarantee

In TSN, low latency guarantees are typically achieved through well-designed flow control, which includes traffic shaping and flow scheduling. Traffic shaping relies on various TSN shapers, each defining the traffic forwarding mechanism on TSN switches. Flow scheduling generates a network-wide schedule deployed on each device, specifying the timing of every transmitted frame. Building on the various TSN shapers introduced in Section 3.2, this section focuses on discussing the key challenges associated with each TSN shaper.

5.2.1 IEEE 802.1Qbv. Although the key idea of IEEE 802.1Qbv Time-Aware Shaper (TAS) mechanism is rather simple, there is an inherent complexity in generating the GCLs, i.e., deciding the right time instances to open and close the gates. This complexity is due to the NP-completeness of the TSN scheduling problem [74], and thus, no polynomial time scheduling algorithm exists unless $P = NP$. To this end, many TAS-based scheduling methods have been developed, and these solutions can be classified into two categories. The first class aims to construct specialized search algorithms, i.e., by developing heuristics, meta-heuristics, or genetic algorithms (e.g., **ant colony optimization (ACO)** [49] and meta-heuristics search algorithms [7]). The second class leverages general-purpose tools, such as **integer linear programming (ILP)** [87] or **satisfiability modulo theories (SMT)** solvers [36] to find the exact solutions.

The primary challenge of generating TAS-based schedules is how to manage the trade-off between efficiency and precision. This trade-off arises from two main considerations. First, the choice of scheduling models – such as whether to allow flow preemption, frame fragmentation, and whether to generate the schedule and routing path jointly – impacts this balance. Using a more complex scheduling model, i.e., enabling the above options, can theoretically enhance system schedulability (i.e., the number of scheduled flows in the system) since it provides a larger search space. However, this also incurs higher computational overhead, which can be counterproductive in practice, especially in resource-constrained systems where a feasible schedule cannot be found by the algorithm in a reasonable amount of time. Another consideration for the trade-off is the choice of scheduling method category, i.e., heuristics or exact solutions. Specifically, heuristic algorithms demonstrate higher efficiency, particularly in large-scale networks, but they may not be able to find any feasible schedule in many cases. On the other hand, an exact algorithm can always find a feasible solution (if it exists) to exhibit superior schedulability performance in small-scale networks.

Besides the precise configuration of switches, the TAS shaper imposes high performance requirements on end stations where it requires the co-design of TSN end stations and gate scheduling on switches to schedule the e2e frame transmissions. Many commercial TSN switch products (e.g., TTTech Evaluation Board [64] and Cisco Industrial Ethernet 4000 Switch [35]) can support real-time and high-throughput (e.g., 1 Gbps) traffic with microseconds-level precision. However, the design of real-time TSN-compatible end station is much more challenging and remains an open problem [68, 153]. Another notable challenge of TAS-based scheduling is the co-scheduling of time-triggered (TT) traffic and synchronization traffic. If transmission collision between the two traffic types occurs, it can cause synchronization error out of bound, resulting in network failure or deadline miss of TT traffic.

5.2.2 IEEE 802.1Qbu. IEEE 802.1Qbu Frame Preemption is beneficial to achieve bounded low latency, especially for critical traffic by preempting the transmission of non-critical traffic. The standard, however, only defines a one-level frame preemption paradigm where frames are classified into express frames or preemptable frames, depending on the criticality of the frames. While one-level preemption can ensure the transmission of high-priority critical traffic to some extent and is relatively simple to implement, it suffers from low flexibility since frames of the same category cannot preempt each other. To address this issue, some studies (e.g., [91]) have proposed the concept of multi-level preemption. By introducing more frame categories, multi-level preemption allows for finer-grained preemption between frames. This approach enhances flexibility and can more effectively reduce frame latency. However, it also significantly increases the configuration complexity. For applications requiring deterministic real-time performance, the worst-case analysis of a multi-level preemption TSN network becomes highly complicated.

TSN supports the concurrent operation of multiple shapers (e.g., TAS and CBS) on the same egress port, and thus utilizing frame preemption in such complex TSN setups can bring many benefits [39]. However, considering that the generation of the GCL is already an NP-hard problem, as described in Section 5.2.1, the use of frame preemption on combined TSN shapers would further elevate the difficulty and complexity of the configuration. Without highly effective and efficient traffic scheduling and configuration methods, combining so many functions could have adverse effects, such as incorrect configurations that fail to ensure timing correctness [12].

Since each occurrence of preemption divides the frame transmission into more segments, additional context switching is required. Therefore, the overhead introduced by preemption is another crucial consideration. Specifically, each preemption incurs a fixed overhead of 12 bytes, as well as the **InterFrame Gap (IFG)** of 12 bytes required between two consecutive transmissions [90]. Moreover, when considering multi-level preemption, each preemption level introduces additional hardware implementation overheads. Thus, although the benefits of preemption are evident, addressing the trade-off between the performance gains from frame preemption and the associated overhead presents a significant challenge.

5.2.3 Other Shapers. The CBS shaper avoids starvation for best-effort flows at the expense of the transmission delay of higher priority and presumably more critical flows [19]. Although CBS is straightforward to implement, networks applying CBS are complex in analyzing the timing performance. In addition, TSN networks with high-volume traffic may suffer from poor performance under CBS in terms of delay guarantee [118]. The PS shaper coordinates operations for both enqueue and dequeue processes, ensuring that all frames are transmitted exactly within their designated time slots. This strict timing requirement means that PS shapers necessitate precise alignment of cycle times, making them less adaptable to asynchronous networks. On the other hand, the ATS shaper aims to achieve bounded low latency for mixed-type traffic without global time synchronization. ATS provides less determinism for critical traffic than TAS but ensures a better average latency of all streams, as evaluated in [173]. However, the current formula of ATS delay bound is rather conservative, where more precise timing analysis is required.

While TSN defines various shapers that can provide real-time deterministic performance for critical traffic, this is usually based on the assumption of a homogeneous network where all devices support these shapers, and there is global network time synchronization. However, industrial automation systems typically include a variety of devices, e.g., PLCs and other legacy equipment. TSN's vendor-independent interoperability feature allows for the existence of such heterogeneous networks within industrial systems. In heterogeneous networks with unscheduled and/or unsynchronized devices, meeting timing requirements remains a significant challenge. Designing effective scheduling mechanisms and timing analysis methods is essential to address this issue. These mechanisms need to ensure that even in the presence of diverse device capabilities and synchronization states, the network can still meet the stringent timing requirements of critical traffic [17].

5.3 Reliability

TSN enhances the reliability of industrial networks through several standardization efforts, including IEEE 802.1CB, IEEE 802.1Qca, and IEEE 802.1Qci, as described in Section 3.3. However, these standards do not specify the exact implementation methods, leaving many research questions on fault tolerance to improve TSN reliability. In general, enhancing TSN reliability involves providing transmission redundancy, at both space and time dimensions.

TSN standards typically use space redundancy. Specifically, IEEE 802.1Qca allows the creation of multiple paths between talkers and listeners for communication, while IEEE 802.1CB defines how to send duplicate traffic frames over different paths and eliminate redundant copies at the destination. This approach is well-suited for handling permanent faults, such as link breaks. The number of faults that can be tolerated depends on the number of redundant paths created [9]. However, space redundancy consumes significant network resources since the redundant paths are typically pre-established with bandwidth pre-allocated, regardless of whether faults occur during the operation. In addition, configuring multiple redundant paths and frame copies increases the complexity of network scheduling.

In contrast, time redundancy based on retransmission is more cost-effective. It creates multiple redundant copies of individual frames over time for retransmission. Unlike space redundancy, time redundancy is better suited for handling transient faults, e.g., packet loss and data error, which may result in incorrect reception and compromised data integrity [162]. The efficiency of time redundancy is also evident in its ability to differentiate the fault probabilities between different links. Indeed, the possibility of faults varies among links due to their physical characteristics. Therefore, time redundancy can allocate a different number of retransmissions for transmissions over different hops based on this information. Research in this area primarily focuses on how to meet reliability requirements, e.g., transmission success rates, with the minimum number of retransmissions [48].

However, both space redundancy and time redundancy methods introduce additional network resource overhead, inevitably impacting other system performance, e.g., schedulability. To further improve resource utilization, adopting resource-sharing methods to provide redundancy is also effective [80, 83]. For example, in space redundancy methods, multiple paths can share one or more links, where partially disjoint paths can result in duplicate frames at intersection switches. In time redundancy methods, multiple traffic flows can share some time slots for retransmissions [166]. However, these resource sharing methods must involve precise analysis of transmission success probabilities by considering various potential transmission scenarios, which poses a great research challenge. An alternative approach to avoiding these highly complex analyses is to use learning-based methods, e.g., federated learning [48], to protect a network with probabilistic link failures.

It is also crucial to make TSN resilient to adversarial attacks. TSN addresses this by defining IEEE 802.1Qci, which provides QoS protection through traffic suppression and blocking. 802.1Qci performs per-stream filtering and policing to protect against unnecessary bandwidth consumption, burst sizes, and malicious or improperly configured endpoints [151]. It can also be used to confine network faults to specific areas, minimizing the impact on other parts of the network [78]. Although 802.1Qci is a published standard, there has been little research on deploying the standard on industrial network devices. One major challenge is how to configure the policing and filtering mechanisms of 802.1Qci, as misconfigurations can result in legitimate packets being filtered out or malicious packets being forwarded [44], which degrades the network reliability and resilience.

5.4 Resource Management

Resource management is essential for provisioning and managing network resources in TSN. It can significantly impact network performance across various aspects, including network deployment, network configuration, traffic scheduling/routing, fault recovery, and network security. TSN primarily relies on the IEEE 802.1Qcc standard for resource management, complemented by the YANG model defined in IEEE 802.1Qcp, which provides a unified data template for network device configuration.

802.1Qcc provides a set of tools for globally managing and reconfiguring the network, specifying three configuration models with regards to their architecture, as described in Section 3.4.2. In general, each model[7] has its strengths and weaknesses, and no single model is applicable to all industrial scenarios [115]. The centralized model controls and manages traffic flows across the entire network, offering precise configuration and reconfiguration to meet timing and reliability requirements due to its global network knowledge [165]. However, this model has several flaws. The reliance on a single centralized controller makes the network vulnerable; if the controller fails, the network must maintain its current configuration and operating status until the controller is restored, rendering it unable to respond to network dynamics (e.g., adding new traffic) or

[7] Since both the fully centralized model and the centralized network/distributed user model utilize CNC to configure TSN elements, we refer to them as the centralized model.

failures. In addition, centralized models suffer from poor scalability. In large-scale networks, their response times can be considerably large due to reliance on the CNC and multicast broadcasting mechanisms to handle various network dynamics [164]. Furthermore, since a large amount of the computational workload is concentrated on the centralized controller, its computational performance can become a bottleneck for the entire network. On the other hand, the distributed model avoids the added complexity and single point of failure associated with centralized management and provides a much faster response to network dynamics since it does not require extensive configuration information exchange across the entire network. However, compared to centralized methods, it has slow network convergence and may result in transmission collisions, thus falling short of the network performance compared to those achieved by centralized methods. Therefore, selecting the appropriate resource management model and specific configuration methods based on the particular industrial application scenario and the corresponding application QoS requirements is a significant challenge. This decision must balance the trade-offs between complexity, responsiveness, scalability, and performance to ensure optimal network operation tailored to the unique demands of each industrial setting.

Although IEEE 802.1Qcc is a published standard, the specified functions of the introduced CNC and CUC are not clearly defined. The implementation of the communication interface UNI between these TSN elements also needs further study. To this end, an ongoing standard, IEEE P802.1Qdj [2], specifies enhancements to the UNI to include new capabilities to support bridges and end stations to extend the configuration capability. It also clarifies the functions of CNC and CUC, and stipulates the YANG model used for the communication between CNC and CUC. However, there is very limited research on these standards, leaving many challenging issues to be studied, e.g., the selection of appropriate resource management protocol among many candidates, including NETCONF, CORECONF, and RESTCONF [21].

Furthermore, enabling efficient and effective network reconfiguration in response to various TSN network dynamics is a challenging task. For efficiency, industrial automation requires on-the-fly control and configuration to handle network dynamics without causing system downtime [33]. This requires to avoid complex reconfiguration algorithms, e.g., SMT-based solutions, which require a long time to solve. For effectiveness, online reconfiguration must still meet stringent QoS requirements, particularly timing guarantees for critical traffic, even during dynamic adjustments. In this regard, centralized methods have their advantage since they have global network information. However, given the complexity of GCL configuration and routing determination, this remains a highly challenging problem.

Industrial automation systems may involve legacy or off-the-shelf end systems (e.g., PLC) that are unscheduled and/or unsynchronized. Dynamic reconfiguration for such heterogeneous TSN networks introduces another level of complexity since the TSN flows need to pass through the non-TSN network [103]. This brings significant uncertainty to latency and jitter, requiring precise timing analysis to preserve the determinism of critical flows.

6 Research Directions

In this section, we discuss several future research directions of TSN, including real-world field deployment, large-scale industrial network design, and wireless TSN. We believe that R&D efforts in these areas will further support the seamless integration of TSN into industrial automation.

6.1 TSN Deployment

The TSN standards are still work in progress and require substantial modification, testing, and validation before wide deployment in real fields. In the following, we discuss several open R&D problems related to TSN deployment and outline the future directions.

6.1.1 Configuration Synthesis. Given the network configuration and application requirements, the system designer needs to solve the so-called network-wide configuration synthesis problem [41], i.e., determining the set of combined mechanisms that can satisfy the application requirements. Configuration synthesis is critical for industrial automation as different applications may have specific functional requirements. To maximize the benefits of applying TSN in the automation industry, the system designer must clearly understand the required functionality and make trade-offs in selecting specific TSN standards. The effects of using various standards in combination can lead to complex network configurations, potentially hindering the full utilization of TSN capabilities in industrial automation systems. This may further introduce extra costs during the product's lifetime if the selected technology needs replacement during or after deployment. Changing the selected standards would require significant redesign, installation, and re-verification [59].

6.1.2 Coexistence of Shapers. With the advancement of industrial automation, many emerging industrial applications often have diverse QoS requirements. This requires TSN to support a range of time-sensitive applications by combining different shapers. This motivates an important future research direction to study the benefits and pitfalls of the coexistence of different types of shapers in the system. Some studies have already explored shaper combinations such as TAS + CBS (e.g., [20, 62]) and TAS + CQF (e.g., [89, 143, 160]). When multiple shapers coexist in a system, they may interact with each other, potentially affecting overall performance. How to ensure that the key characteristics of TSN, especially e2e timing analysis, are maintained under these conditions deserves further investigation.

6.1.3 Dynamic Reconfiguration. Industrial applications may suffer from unexpected dynamics (e.g., network topology updates and traffic specification changes) during the network operation. This requires dynamic TSN reconfiguration by adding, removing, or changing network devices and application tasks flexibly at run time. Although offline TSN configuration enables precise construction of communication schedules to provide deterministic performance for real-time industrial applications, it does not allow flexible network reconfiguration. To enable efficient and effective online reconfiguration, it requires a deep understanding of the dynamic configuration process, especially the associated timing overhead in each reconfiguration [94]. Then, effective dynamic reconfiguration methods based on different mechanisms (e.g., incremental reconfiguration [53] or pre-allocated partition [141]) should be further explored.

6.1.4 Security. Security is always a critical concern in industrial automation, and ensuring TSN security remains an open research topic. The IEEE 802.1 Security TG, part of the IEEE 802.1 WG, is actively working on enhancing TSN's secure capabilities, with ongoing cooperation between the IEEE 802.1 TSN TG and the IEEE 802.1 Security TG. However, as the automation industry becomes more open to the public, TSN-enabled systems will be exposed to various existing and novel attacks. Further research in TSN security is highly needed for early detection of these threats and development of effective mitigation strategies [18].

6.2 Large-Scale Industrial Networks

In the current practice, TSN is mainly deployed in relatively small-scale LANs, enabling the connection among floor shop devices in factory-size networks. The maximum e2e latency of time-sensitive traffic classes can only be guaranteed up to seven hops, which significantly limits TSN's scalability [130].

6.2.1 DetNet. To improve the scalability of TSN, the IETF DetNet group is working in collaboration with the TSN TG to develop standardization of IP layer deterministic forwarding services applied to Layer 3 routed segments. TSN/DetNet integration facilitates transforming isolated local real-time networks into integrated large-scale networks. Although DetNet standards are still under development, extensive research (e.g., [136, 150]) has been conducted based on **Request for Comments (RFC)** documents [137] and technical guidance drafts. However, research on DetNet over TSN is still at its initial stage, especially for deployment in large-scale industrial networks spanning large geographic areas. Ensuring consistent QoS performance (espeically for the timing guarantees) for such cross-network real-time communication poses many challenges. For example, long propagation delays between adjacent switches along a multi-hop path in a large-scale network can introduce significant jitter and reduce synchronization precision. Additionally, traffic scheduling in a cross-network setting becomes more complex as relying on a centralized controller (i.e., CNC) to pre-compute the network-wide schedule is not feasible anymore. Exploring distributed (e.g., [113]) or hierarchical scheduling mechanisms (e.g., [142]) could lead to be possible solutions.

6.2.2 Virtualization. A large-scale industrial automation system is typically an integration of heterogeneous computing and communication platforms containing diverse hardware, e.g., multicore CPUs, GPUs, MCUs, and FPGAs. The stringent timing requirements further drives the industrial automation systems to employ the edge-cloud computing paradigm with a hierarchy of computing resources. To manage these heterogeneous resources, resource virtualization is an enabling technique that can help reduce the operation expenses and increase the system flexibility and scalability since applications running on **virtual machines (VMs)** can be easily managed (e.g., create, migrate, or delete) [140]. However, the use of TSN in virtual environments is a relatively new trend as the TSN standards were originally intended for bare-metal industrial applications and recently there have been some pioneering work on this topic (e.g., architecture hypotheses [71] and testbed validation [52]). Despite the potential advantages provided by resource virtualization, it is still an open research problem with many challenges unsolved. First, virtualization may introduce a source of unpredictability (e.g., unpredictable latency caused by VMs running on adjacent cores) that may lead to the loss of determinism. To achieve the desired flexibility, VM placement and dynamic VM migration (e.g., virtual PLCs) pose challenges in online TSN scheduling in response to dynamic changes of application requirements. In addition, to mitigate any form of overhead, lightweight virtualization techniques have become the standard technology for edge components, e.g., using containerization instead of hypervisor-based VMs [51]. The highly distributed nature of edge cloud applications is a challenge to effectively supporting the most performance-demanding components in containerization frameworks.

6.3 Wireless TSN

Most existing industrial automation systems rely on Ethernet-based fieldbus communication, which are based on wired connections. Applying wireless technologies to the automation industry provides many obvious advantages, e.g., reduced wiring cost and improved device mobility. Many industrial automation use cases can directly benefit from TSN capabilities over wireless, e.g., closed loop control, mobile robots, and autonomous ground vehicles [29]. However, given the inherently unreliable characteristic of wireless connection, achieving wireless TSN is challenging [31], particularly in providing deterministic timing and reliability guarantees. Wireless media has fundamental differences from their wired counterparts, e.g., varied transmission capacity depending on link quality and unreliable nature due to stochastic properties of the channel and interference. These challenges motivate a number of future research topics.

6.3.1 Time Synchronization. Both industry and academia have been actively working on the design and development of wireless TSN, where IEEE 802.11 and 5G are considered the two major candidates. For this aim, achieving accurate time synchronization is the first step towards making TSN available on wireless networks, and it is the foundation for time-critical traffic scheduling to achieve deterministic real-time communication. Different from wired industrial networks, time synchronization over wireless networks needs to tackle several challenges (e.g., high delay variation and imprecise timestamping), and there is a rich literature on analyzing or providing real-world implementations of the integration of wired and wireless clock synchronization for both IEEE 802.11 and 5G.

For IEEE 802.11, there are mainly three messaging schemes to perform clock synchronization: (1) IEEE 802.1AS messaging relying on the de facto PTP standard [57, 134], (2) IEEE 802.11 messaging by integrating **Fine Timing Measurement (FTM)** into 802.1AS [126], and (3) low-overhead beacon-based time synchronization mechanism [61, 104].

For 5G, the clock synchronization support is standardized in the **Third Generation Partnership Project (3GPP)** Release 16 [3] and mainly two time synchronization approaches are considered [121]: boundary clock and transparent clock. The former requires the 5G **Radio Access Network (RAN)** to have a direct connection to the TSN master clock based on IEEE 802.1AS [116]. The latter is achieved via PTP messages among any forwarding devices by passing relevant time event messages [84, 122]. While the boundary clock approach is simpler to implement, the transparent clock approach is mostly preferred due to its much higher accuracy [105]. Despite significant research progress, time synchronization for wireless TSN still faces many challenges deserving further investigation, including the lack of hardware-timestamping, synchronization errors during handover, and asymmetry in uplink/downlink propagation delay which adversely affect the synchronization process.

6.3.2 Traffic Scheduling. To meet deterministic timing guarantees in wireless TSN, besides precise time synchronization, another critical research area is timing-aware traffic scheduling. IEEE 802.11's default medium access is contention-based and non-deterministic. Thus, a significant amount of research has explored replacing/improving traditional 802.11 MAC with TDMA-based MAC protocols (e.g., [111, 146, 158]). Other efforts have focused on implementing 802.1Qbv on the network stack using TSN functionalities and tools available in the Linux kernel (e.g., [124]). In the meantime, IEEE 802.11 is rapidly evolving to support time-sensitive applications in industrial automation. For example, Wi-Fi 6 (802.11ax) supports several methods (e.g., the scheduled **trigger frame (TF)**-based access scheme) to enable wireless TSN-capable **access points (APs)** and to ensure nearly deterministic transmissions. Further enhancement to the 802.11ax TF is also under consideration by Wi-Fi 7 (802.11be) to deterministically schedule 802.11 frames [5]. It is expected that more research will emerge to address other open challenges, e.g., supporting ultra-low latency and frame preemption [30].

5G, as another wireless TSN candidate, does not share the same IEEE 802-based link layer as Ethernet and Wi-Fi, while 5G-TSN integration is also feasible via translation interfaces defined in 3GPP Rel. 16 [3]. 5G can be integrated within the TSN network as a logical TSN bridge where the 5G core and RAN remain hidden from the TSN network. To inter-operate between TSN and 5G systems, 3GPP introduces the TSN translator functionality at the interconnection points between both networks. The translator functionality, both in the device side and the network side acting as TSN ingress and egress ports, is to configure all parameters necessary to coordinate 5G and TSN [147]. These translators realize the configuration of the 5G system in order to fulfill the required TSN deterministic transmissions with bounded latency. 5G **ultra-reliable low-latency communications (URLLC)** provide a good match to TSN features by enabling increased reliability and latency below 1 *ms*. Significant research works (e.g., [54, 95, 114, 167, 168]) have also studied the real-time scheduling problems of URLLC traffic in industrial applications to meet their stringent timing requirements. These solutions, however, are more suitable in standalone industrial 5G networks instead of 5G-TSN integration systems which must follow the schedule specified by CNC in TSN. To achieve this, internal configuration is required for the 5G system, including mapping traffic classes in TSN into a predefined **5G QoS indicator (5QI)** and leveraging hold & forward buffering mechanism which is identical to the gate scheduling behavior of TSN GCL [67]. Although 3GPP specification provides a comprehensive mapping from 5G to TSN traffic shaping and scheduling, the wireless nature that allows mobility and frequent changes in the network layout requires further enhancements to the traffic scheduling mechanism design.

6.3.3 Reliability. In addition to time synchronization and traffic scheduling, guaranteeing the reliability of transmissions is another key challenge to enable wireless TSN. The ultra-reliability feature in wireless networks is typically pursued through enabling transmission redundancy in different manners including (1) intra-frame redundancy, (2) inter-frame redundancy, and (3) multi-path redundancy. Intra-frame redundancy introduces redundant bits within a frame to increase the probability of successful reception of a frame. 802.11 and 5G both support intra-frame redundancy via the configuration of **modulation and coding scheme (MCS)** specifying the ratio of redundant bits in a frame. Inter-frame redundancy performs frame retransmissions either actively (i.e., after detecting transmission failure through ACK) or passively (i.e., reserving multiple frame copies). Active redundancy is spectrum-efficient but suffers from higher transmission latency. Thus, passive redundancy is a compelling method to achieve ultra-reliability in wireless TSN without sacrificing latency. In multi-path redundancy, multiple copies of a frame are transmitted to the destination through different paths or links. 802.11 supports multi-link operation allowing a station to simultaneously maintain multiple 802.11 links across the 2.4, 5, and 6 GHz bands. 5G can enable multi-path redundancy through setting up redundant **Protocol Data Unit (PDU)** sessions where different solutions can be applied [13]. Currently, the transmission interference is still a major hurdle to achieving ultra-high reliability, especially for communications in unlicensed bands like Wi-Fi. Power management is another direction since an increment of the transmission power improves the transmission reliability but may decrease the power efficiency of the wireless system.

6.3.4 Wireless Security. Providing security and safety guarantees is critical for industrial automation systems. TSN defines the 802.1Qci protocol to block malicious devices or attacks and 802.1Qci provides traffic filtering and policing schemes at the ingress port of switch to prevent unidentified traffic, thereby improving network security. Many researchers also discuss the design of fault detection methods and encryption mechanisms based on 802.1Qci (e.g., [78, 144]) to further enhance the network security. Many other strategies (e.g., authentication, encryption and decryption, intrusion prevention) may also be deployed to achieve e2e security in TSN. However, the trade-off between the cyber security and TSN performance must be considered since cyber security strategies can introduce additional traffic transmission delay which further impact the determinism of the network.

Comparing Ethernet-based TSN networks with wireless TSN networks, they share similar security objectives at a high level, but wireless networks are more vulnerable to attacks, e.g., eavesdropping and tampering [88]. To address these security concerns, **Wi-Fi Protected Access (WPA)** is an authentication and key management protocol developed for encryption in Wi-Fi and WPA2 is retired by the new standard WPA3 to make Wi-Fi more secure [66]. For 5G, 3GPP defines several security domains, e.g., network domain security, user domain security, and application domain security, with many solutions standardized throughout the evolution of cellular technologies, including mutual authentication and authorization of the network and the UE, integrity protection of the RRC-signaling and NAS-signaling, and so on [4]. Additionally, in the context of 5G-TSN deployment, unique challenges such as clock skew in GM-based time synchronization, **denial of service (DoS)** attack, and **rogue base station (RBS)** should also be investigated [112].

In summary, supporting wireless TSN requires careful selection of design approaches, considering several trade-offs in the design process. These include the trade-off between scheduling complexity and handover delay in dealing with user mobility [14], the trade-off between deterministic performance guarantee and associated radio resource costs [75], and the trade-off between the reliability and traffic aggregation overhead [161]. Based on the development of wireless TSN, the next research step is clear that wireless TSN and wired TSN must be integrated to create hybrid TSN networks [110]. The integration of the technologies poses several challenges. Essentially, a hybrid TSN network must maintain the TSN features across the different communication domains and technologies, including guaranteed e2e latency, clock synchronization, and coexistence of traffic flows with different criticality requirements.

7 Conclusion

The industrial automation market is still dominated by Ethernet-based fieldbus systems, particularly those with real-time capabilities, e.g., EtherCAT, PROFINET IRT, POWERLINK, and SERCOS III. Although these technologies are based on conventional Ethernet, they are not designed to interoperate with fieldbus from other vendors. In the context of industrial automation, a large number of vendor-crossing devices with diverse QoS requirements are expected to communicate across all levels of the automation pyramid. Thus, TSN has the potential to enable modern industrial automation by establishing universal physical and data-link layer standards. TSN consists of a set of Ethernet-based protocols and standards designed to address a wide range of practical industrial use cases with guaranteed timing requirements in heterogeneous networks. TSN encompasses a broad scope, making it critical to understand the standards systematically rather than focusing on just one characteristic or component. This paper provides a comprehensive review of TSN standards in industrial automation, including both published standards and in-progress drafts. We specifically focus on the automation industry, discussing the challenges and opportunities when applying TSN to industrial control applications. In addition, we highlight promising research directions for TSN design and development in industrial automation, such as optimizing current TSN standards and integrating TSN with other technologies.

References

[1] 2024. Draft standard for local and metropolitan area networks: Timing and synchronization for time-sensitive applications – amendment: Hot standby and clock drift error reduction. *IEEE P802.1ASdm Draft 2.4, 2024* (June 2024).

[2] 2024. IEEE standard for local and metropolitan area networks–bridges and bridged networks amendment 38: Configuration enhancements for time-sensitive networking. *IEEE Std 802.1Qdj-2024 (Amendment to IEEE Std 802.1Q-2022 as amended by IEEE Std 802.1Qcz-2023, IEEE Std 802.1Qcw-2023, and IEEE Std 802.1Qcj-2023)* (2024), 1–56. https://doi.org/10.1109/IEEESTD.2024.10542670

[3] 3GPP. 2020. 5G; system architecture for the 5G system (5GS). *3rd Generation Partnership Project (3GPP), Technical Report (TR) 23.501* (2020). Version 16.5.1.

[4] 3GPP. 2024. Security architecture and procedures for 5G System. *3rd Generation Partnership Project (3GPP), Technical Specification (TS) 33.501* (03 2024). Version 18.5.0.

[5] Toni Adame, Marc Carrascosa-Zamacois, and Boris Bellalta. 2021. Time-sensitive networking in IEEE 802.11 be: On the way to low-latency WiFi 7. *Sensors* 21, 15 (2021), 4954.

[6] A. Ademaj, D. Puffer, D. Bruckner, G. Ditzel, L. Leurs, M. P. Stanica, P. Didier, R. Hummen, R. Blair, and T. Enzinger. 2019. Industrial automation traffic types and their mapping to QoS/TSN mechanisms. *TSN Mechanisms* (2019), 3.

[7] M. Adly and A. H. Besheer. 2013. A meta-heuristics search algorithm as a solution for energy transfer maximization in stand-alone photovoltaic systems. *International Journal of Electrical Power & Energy Systems* 51 (2013), 243–254.

[8] Mohamad Kenan Al-Hares, Philippos Assimakopoulos, Daniel Muench, and Nathan J. Gomes. 2017. Modeling time aware shaping in an Ethernet fronthaul. In *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 1–6.

[9] Inés Álvarez, Drago Čavka, Julián Proenza, and Manuel Barranco. 2019. Simulation of the proactive transmission of replicated frames mechanism over TSN. In *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'19)*. IEEE, 1375–1378.

[10] Anna Arestova, Kai-Steffen Jens Hielscher, and Reinhard German. 2021. Simulative evaluation of the TSN mechanisms time-aware shaper and frame preemption and their suitability for industrial use cases. In *2021 IFIP Networking Conference (IFIP Networking)*. IEEE, 1–6.

[11] Mohammad Ashjaei, Lucia Lo Bello, Masoud Daneshtalab, Gaetano Patti, Sergio Saponara, and Saad Mubeen. 2021. Time-Sensitive Networking in automotive embedded systems: State of the art and research opportunities. *Journal of Systems Architecture* 117 (2021), 102137.

[12] Mohammad Ashjaei, Lejla Murselović, and Saad Mubeen. 2022. Implications of various preemption configurations in TSN networks. *IEEE Embedded Systems Letters* 14, 1 (2022), 39–42.

[13] Mahin K. Atiq, Raheeb Muzaffar, Óscar Seijo, Iñaki Val, and Hans-Peter Bernhard. 2022. When IEEE 802.11 and 5G meet time-sensitive networking. *IEEE Open Journal of the Industrial Electronics Society* 3 (2022), 14–36.

[14] Pablo Avila-Campos, Jetmir Haxhibeqiri, Xianjun Jiao, Ben Van Herbruggen, Ingrid Moerman, and Jeroen Hoebeke. 2024. Unlocking mobility for Wi-Fi-based wireless time-sensitive networks. *IEEE Access* (2024).

[15] Mohd Aiman Kamarul Bahrin, Mohd Fauzi Othman, NH Nor Azli, and Muhamad Farihin Talib. 2016. Industry 4.0: A review on industrial automation and robotic. *Jurnal Teknologi* 78, 6-13 (2016), 137–143.

[16] Balakrishna Balakrishna, Boris Meinardus, and Leonidas Kontopoulos. 2021. Simulation framework for EtherCAT over TSN. In *2021 IFIP Networking Conference (IFIP Networking)*. IEEE, 1–6.

[17] Mohammadreza Barzegaran, Niklas Reusch, Luxi Zhao, Silviu S. Craciunas, and Paul Pop. 2022. Real-time traffic guarantees in heterogeneous time-sensitive networks. In *Proceedings of the 30th International Conference on Real-Time Networks and Systems*. 46–57.

[18] Lucia Lo Bello and Wilfried Steiner. 2019. A perspective on IEEE time-sensitive networking for industrial communication and automation systems. *Proc. IEEE* 107, 6 (2019), 1094–1120.

[19] Nassima Benammar, Henri Bauer, Frédéric Ridouard, and Pascal Richard. 2018. Timing analysis of AVB Ethernet network using the forward end-to-end delay analysis. In *Proceedings of the 26th International Conference on Real-Time Networks and Systems*. 223–233.

[20] Daniel Bezerra, Assis T. de Oliveira Filho, Iago Richard Rodrigues, Marrone Dantas, Gibson Barbosa, Ricardo Souza, Judith Kelner, and Djamel Sadok. 2022. A machine learning-based optimization for end-to-end latency in TSN networks. *Computer Communications* 195 (2022), 424–440.

[21] Mahesh Ganesh Bhat, Sushmit Bhattacharjee, Cenk Gündoğan, Konstantinos Alexandris, and Alexander Gogolev. 2023. CORECONF, NETCONF, and RESTCONF: Benchmarking network orchestration in constrained IIoT devices. *IEEE Internet of Things Journal* (2023).

[22] Martin Björklund. 2016. The YANG 1.1 Data Modeling Language. RFC 7950. https://doi.org/10.17487/RFC7950

[23] Stuart A. Boyer. 2009. *SCADA: Supervisory Control and Data Acquisition*. International Society of Automation.

[24] Hugh Boyes, Bil Hallaq, Joe Cunningham, and Tim Watson. 2018. The industrial internet of things (IIoT): An analysis framework. *Computers in Industry* 101 (2018), 1–12.

[25] Simon Brooks and Ecehan Uludag. 2018. *Time-Sensitive Networking: From Theory to Implementation in Industrial Automation*. TTTech. White Paper.

[26] Dietmar Bruckner, Richard Blair, M. Stanica, A. Ademaj, W. Skeffington, D. Kutscher, S. Schriegel, R. Wilmes, K. Wachswender, L. Leurs, M. Seewald, R. Hummen, E.-C. Liu, and S. Ravikumar. 2018. Opc ua tsn a new solution for industrial communication. *Whitepaper. Shaper Group* 168 (2018), 1–10.

[27] Dietmar Bruckner, Marius-Petru Stănică, Richard Blair, Sebastian Schriegel, Stephan Kehrer, Maik Seewald, and Thilo Sauter. 2019. An introduction to OPC UA TSN for industrial communication systems. *Proc. IEEE* 107, 6 (2019), 1121–1131.

[28] Thomas Burns, John Cosgrove, and Frank Doyle. 2019. A review of interoperability standards for Industry 4.0. *Procedia Manufacturing* 38 (2019), 646–653.

[29] Dave Cavalcanti, Stephen Bush, Marc Illouz, Genio Kronauer, Alon Regev, and Ganesh Venkatesan. 2020. Wireless TSN: Definitions, use cases & standards roadmap. *Avnu Alliance* (2020), 1–16.

[30] Dave Cavalcanti, Carlos Cordeiro, Malcolm Smith, and Alon Regev. 2022. WiFi TSN: Enabling deterministic wireless connectivity over 802.11. *IEEE Communications Standards Magazine* 6, 4 (2022), 22–29.

[31] Dave Cavalcanti, Javier Perez-Ramirez, Mohammad Mamunur Rashid, Juan Fang, Mikhail Galeev, and Kevin B. Stanton. 2019. Extending accurate time distribution and timeliness capabilities over the air to enable future wireless industrial automation systems. *Proc. IEEE* 107, 6 (2019), 1132–1152.

[32] CC-Link Partner Association (CLPA) 2023. *CC-link IE TSN Compatible Product Development Method Guide*. CC-Link Partner Association (CLPA). White Paper.

[33] Hamza Chahed and Andreas J. Kassler. 2021. Software-defined time sensitive networks configuration and management. In *2021 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN'21)*. IEEE, 124–128.

[34] Yingxiu Chen, Qimin Xu, Jinglong Zhang, Lei Xu, Lingzhi Li, and Cailian Chen. 2022. TSN-compatible industrial wired/wireless multi-protocol conversion mechanism and module. In *IECON 2022–48th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 1–6.

[35] Cisco. 2023. Industrial Ethernet 4000 Series Switches - Data Sheet.

[36] Silviu S. Craciunas and Ramon Serna Oliver. 2014. SMT-based task-and network-level static schedule generation for time-triggered networked systems. In *Proceedings of the 22nd International Conference on Real-Time Networks and Systems*. 45–54.

[37] Peter Danielis, Henning Puttnies, Eike Schweissguth, and Dirk Timmermann. 2018. Real-time capable internet technologies for wired communication in the industrial IoT-a survey. In *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA'18)*, Vol. 1. IEEE, 266–273.

[38] Rubi Debnath, Philipp Hortig, Luxi Zhao, and Sebastian Steinhorst. 2023. Advanced modeling and analysis of individual and combined TSN shapers in OMNeT++. In *2023 IEEE 29th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA'23)*. IEEE, 176–185.

[39] Rubi Debnath, Philipp Hortig, Luxi Zhao, and Sebastian Steinhorst. 2024. Quantifying the impact of frame preemption on combined TSN shapers. *arXiv preprint arXiv:2401.13631* (2024).

[40] Paul Didier and Joseph Fontaine. 2017. Results, insights and best practices from IIC testbeds: Time-sensitive networking testbed. *Industrial Internet Consortium* (2017).

[41] Ahmed El-Hassany, Petar Tsankov, Laurent Vanbever, and Martin Vechev. 2017. Network-wide configuration synthesis. In *Computer Aided Verification: 29th International Conference, CAV 2017, Heidelberg, Germany, July 24–28, 2017, Proceedings, Part II 30*. Springer, 261–281.

[42] Vincenzo Eramo, Tiziana Fiori, Francesco G. Lavacca, Francesco Valente, Marta Albano, Simone Ciabuschi, and Enrico Cavallini. 2023. Performance comparisons of flexible time triggered Ethernet and TTEthernet technologies for space launcher networks. In *2023 IEEE 10th International Workshop on Metrology for AeroSpace (MetroAeroSpace'23)*. IEEE, 61–66.

[43] Vincenzo Eramo, Tiziana Fiori, Francesco G. Lavacca, Francesco Valente, Andrea Baiocchi, Simone Ciabuschi, Marta Albano, and Enrico Cavallini. 2023. A max plus algebra based scheduling algorithm for supporting time triggered services in Ethernet networks. *Computer Communications* 198 (2023), 85–97.

[44] Doğanalp Ergenç, Cornelia Brülhart, Jens Neumann, Leo Krüger, and Mathias Fischer. 2021. On the security of IEEE 802.1 time-sensitive networking. In *2021 IEEE International Conference on Communications Workshops (ICC Workshops'21)*. IEEE, 1–6.

[45] Jonathan Falk, David Hellmanns, Ben Carabelli, Naresh Nayak, Frank Dürr, Stephan Kehrer, and Kurt Rothermel. 2019. NeSTiNg: Simulating IEEE time-sensitive networking (TSN) in OMNeT++. In *2019 International Conference on Networked Systems (NetSys'19)*. IEEE, 1–8.

[46] János Farkas. 2018. Introduction to IEEE 802.1: Focus on the Time-Sensitive Networking Task Group. http://www.ieee802.org/1/files/public/docs2018/tsn-farkas-intro-0318-v01.pdf

[47] Janos Farkas, Lucia Lo Bello, and Craig Gunther. 2018. Time-sensitive networking standards. *IEEE Communications Standards Magazine* 2, 2 (2018), 20–21.

[48] Zhiwei Feng, Chaoquan Wu, Qingxu Deng, Meiling Han, and Yuhan Lin. 2023. ReT-FTS: Re-transmission-based fault-tolerant scheduling in TSN. *Journal of Systems Architecture* 142 (2023), 102959.

[49] Fabrizio Ferrandi, Pier Luca Lanzi, Christian Pilato, Donatella Sciuto, and Antonino Tumeo. 2010. Ant colony heuristic for mapping and scheduling tasks and communications on heterogeneous embedded systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 29, 6 (2010), 911–924.

[50] Tiziana Fiori, Francesco G. Lavacca, Francesco Valente, and Vincenzo Eramo. 2024. Proposal and investigation of a lite time sensitive networking solution for the support of real time services in space launcher networks. *IEEE Access* (2024).

[51] Andrea Garbugli, Lorenzo Rosa, Armir Bujari, and Luca Foschini. 2023. KuberneTSN: A deterministic overlay network for time-sensitive containerized environments. In *ICC 2023-IEEE International Conference on Communications*. IEEE, 1494–1499.

[52] Andrea Garbugli, Lorenzo Rosa, Luca Foschini, Antonio Corradi, and Paolo Bellavista. 2022. A framework for TSN-enabled virtual environments for ultra-low latency 5G scenarios. In *ICC 2022-IEEE International Conference on Communications*. IEEE, 5023–5028.

[53] Christoph Gärtner, Amr Rizk, Boris Koldehofe, René Guillaume, Ralf Kundel, and Ralf Steinmetz. 2023. Fast incremental reconfiguration of dynamic time-sensitive networks at runtime. *Computer Networks* 224 (2023), 109606.

[54] David Ginthör, Johannes von Hoyningen-Huene, Rene Guillaume, and Hans Schotten. 2019. Analysis of multi-user scheduling in a TSN-enabled 5G system for industrial applications. In *2019 IEEE International Conference on Industrial Internet (ICII'19)*. IEEE, 190–199.

[55] Alexander Gogolev, Francisco Mendoza, and Rol Braun. 2018. TSN-enabled OPC UA in field devices. In *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA'18)*, Vol. 1. IEEE, 297–303.

[56] Tao Gong, Tianyu Zhang, Xiaobo Sharon Hu, Qingxu Deng, Michael Lemmon, and Song Han. 2019. Reliable dynamic packet scheduling over lossy real-time wireless networks. In *31st Euromicro Conference on Real-Time Systems (ECRTS'19)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

[57] Michael Gundall, Christopher Huber, and Sergiy Melnyk. 2021. Integration of IEEE 802.1 AS-based time synchronization in IEEE 802.11 as an enabler for novel industrial use cases. *arXiv preprint arXiv:2101.02434* (2021).

[58] Wang Guo, Yanhong Huang, Jianqi Shi, Zhe Hou, and Yang Yang. 2021. A formal method for evaluating the performance of TSN traffic shapers using UPPAAL. In *2021 IEEE 46th Conference on Local Computer Networks (LCN'21)*. IEEE, 241–248.

[59] Daniel Hallmans, Mohammad Ashjaei, and Thomas Nolte. 2020. Analysis of the TSN standards for utilization in long-life industrial distributed control systems. In *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'20)*, Vol. 1. IEEE, 190–197.

[60] Mark Hantel and Jordon Woods. 2022. The integration of time-sensitive networking into EtherNet/IP technologies. In *The ODVA 2022 Industry Conference & 21st Annual Meeting*.

[61] Jetmir Haxhibeqiri, Xianjun Jiao, Muhammad Aslam, Ingrid Moerman, and Jeroen Hoebeke. 2021. Enabling TSN over IEEE 802.11: Low-overhead time synchronization for Wi-Fi clients. In *2021 22nd IEEE International Conference on Industrial Technology (ICIT'21)*, Vol. 1. IEEE, 1068–1073.

[62] Bahar Houtan, Mohammad Ashjaei, Masoud Daneshtalab, Mikael Sjödin, and Saad Mubeen. 2024. Bandwidth reservation analysis for schedulability of AVB traffic in TSN. In *2024 IEEE International Conference on Industrial Technology (ICIT'24)*. IEEE, 1–8.

[63] Jahanzaib Imtiaz, Jürgen Jasperneite, and Lixue Han. 2009. A performance study of Ethernet Audio Video Bridging (AVB) for Industrial real-time communication. In *2009 IEEE Conference on Emerging Technologies & Factory Automation*. IEEE, 1–8.

[64] TTTech Industrial. [n. d.]. Edge IP Solution. https://www.tttech-industrial.com/products/slate/edge-ip-solution

[65] Pierre Kehl, Junaid Ansari, Mohammad Hossein Jafari, Paul Becker, Joachim Sachs, Niels König, Amon Göppert, and Robert H. Schmitt. 2022. Prototype of 5G integrated with TSN for edge-controlled mobile robotics. *Electronics* 11, 11 (2022), 1666.

[66] Songhui Kwon and Hyoung-Kee Choi. 2020. Evolution of Wi-Fi protected access: Security challenges. *IEEE Consumer Electronics Magazine* 10, 1 (2020), 74–81.

[67] Ana Larrañaga, M. Carmen Lucas-Estañ, Imanol Martinez, Iñaki Val, and Javier Gozalvez. 2020. Analysis of 5G-TSN integration to support Industry 4.0. In *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'20)*, Vol. 1. IEEE, 1111–1114.

[68] Christoph Lehr. 2023. *Design and Limitations of a Software-based TSN End Station*. Ph. D. Dissertation. Wien.

[69] Paulo Leitão, Armando Walter Colombo, and Stamatis Karnouskos. 2016. Industrial automation based on cyber-physical systems technologies: Prototype implementations and challenges. *Computers in Industry* 81 (2016), 11–25.

[70] Stefan-Helmut Leitner and Wolfgang Mahnke. 2006. OPC UA–service-oriented architecture for industrial applications. (2006).

[71] Luca Leonardi, Lucia Lo Bello, and Gaetano Patti. 2020. Towards time-sensitive networking in heterogeneous platforms with virtualization. In *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'20)*, Vol. 1. IEEE, 1155–1158.

[72] Yuting Li, Junhui Jiang, Changdae Lee, and Seung Ho Hong. 2020. Practical implementation of an OPC UA TSN communication architecture for a manufacturing system. *IEEE Access* 8 (2020), 200100–200111.

[73] Hyung-Taek Lim, Daniel Herrscher, Martin Waltl, and Firas Chaari. 2012. Performance analysis of the IEEE 802.1 Ethernet audio/video bridging standard. In *Fifth International Conference on Simulation Tools and Techniques*.

[74] Yuhan Lin, Xi Jin, Tianyu Zhang, Meiling Han, Nan Guan, and Qingxu Deng. 2021. Queue assignment for fixed-priority real-time flows in time-sensitive networks: Hardness and algorithm. *Journal of Systems Architecture* 116 (2021), 102141.

[75] Yue Liu, Jizhao Lu, Yanru Wang, Hui Liu, Yalin Cao, and Lei Feng. 2023. Multicast wireless resource optimization for high-precision clock synchronization timing service in 5G-TSN. In *International Conference on Computer Engineering and Networks*. Springer, 190–199.

[76] Yinzhi Lu, Guofeng Zhao, Chinmay Chakraborty, Chuan Xu, Liu Yang, and Keping Yu. 2023. Time sensitive networking-driven deterministic low-latency communication for real-time telemedicine and e-health services. *IEEE Transactions on Consumer Electronics* (2023).

[77] Jordon Woods and Ludwig Winkel. 2020. IEC/IEEE 60802 TSN Profile for Industrial Automation. https://1.ieee802.org/tsn/iec-ieee-60802/

[78] Feng Luo, Bowen Wang, Zihao Fang, Zhenyu Yang, and Yifan Jiang. 2021. Security analysis of the TSN backbone architecture and anomaly detection system design based on IEEE 802.1 Qci. *Security and Communication Networks* 2021, 1 (2021), 6902138.

[79] Jin Lv, Yongxin Zhao, Xi Wu, Yongjian Li, and Qiang Wang. 2020. Formal analysis of TSN scheduler for real-time communications. *IEEE Transactions on Reliability* 70, 3 (2020), 1286–1294.

[80] Wenhong Ma, Xiangzhen Xiao, Guoqi Xie, Nan Guan, Yu Jiang, and Wanli Chang. 2023. Fault tolerance in time-sensitive networking with mixed-critical traffic. In *2023 60th ACM/IEEE Design Automation Conference (DAC'23)*. IEEE, 1–6.

[81] Magdi S. Mahmoud. 2019. Architecture for cloud-based industrial automation. In *Third International Congress on Information and Communication Technology*. Springer, 51–62.

[82] Dorin Maxim and Ye-Qiong Song. 2017. Delay analysis of AVB traffic in time-sensitive networks (TSN). In *Proceedings of the 25th International Conference on Real-Time Networks and Systems*. 18–27.

[83] Junhong Min, Woongsoo Kim, Jeongyeup Paek, and Ramesh Govindan. 2023. Effective routing and scheduling strategies for fault-tolerant time-sensitive networking. *IEEE Internet of Things Journal* (2023).

[84] Anas Bin Muslim, Ralf Tönjes, and Thomas Bauschert. 2024. Synchronizing TSN devices via 802.1 AS over 5G networks. *Electronics* 13, 4 (2024), 768.

[85] Ahmed Nasrallah, Akhilesh S. Thyagaturu, Ziyad Alharbi, Cuixiang Wang, Xing Shao, Martin Reisslein, and Hesham Elbakoury. 2019. Performance comparison of IEEE 802.1 TSN Time Aware Shaper (TAS) and Asynchronous Traffic Shaper (ATS). *IEEE Access* 7 (2019), 44165–44181.

[86] Ahmed Nasrallah, Akhilesh S. Thyagaturu, Ziyad Alharbi, Cuixiang Wang, Xing Shao, Martin Reisslein, and Hesham ElBakoury. 2018. Ultra-low latency (ULL) networks: The IEEE TSN and IETF DetNet standards and related 5G ULL research. *IEEE Communications Surveys & Tutorials* 21, 1 (2018), 88–145.

[87] Naresh Ganesh Nayak, Frank Dürr, and Kurt Rothermel. 2016. Time-sensitive software-defined network (TSSDN) for real-time applications. In *Proceedings of the 24th International Conference on Real-Time Networks and Systems*. 193–202.

[88] Rashid Nazir, Asif Ali Laghari, Kamlesh Kumar, Shibin David, and Munwar Ali. 2021. Survey on wireless network security. *Archives of Computational Methods in Engineering* (2021), 1–20.

[89] Hongrui Nie, Yue Su, Weibo Zhao, and Junsheng Mu. 2024. Hybrid traffic scheduling in time-sensitive networking for the support of automotive applications. *IET Communications* 18, 2 (2024), 111–128.

[90] Mubarak Ojewale, Patrick Meumeu Yomsi, and Geoffrey Nelissen. 2018. On multi-level preemption in Ethernet. *Proceedings of the Work-in-Progress Session (ECRTS 2018)* (2018), 16–18.

[91] Mubarak Adetunji Ojewale, Patrick Meumeu Yomsi, and Borislav Nikolić. 2020. Multi-level preemption in TSN: Feasibility and requirements analysis. In *2020 IEEE 23rd International Symposium on Real-Time Distributed Computing (ISORC'20)*. IEEE, 47–55.

[92] Lukas Osswald, Steffen Lindner, Lukas Wüsteney, and Michael Menth. 2021. Rap extensions for the hybrid configuration model. In *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'21)*. IEEE, 1–8.

[93] Özgür Özkaya, Jetmir Haxhibeqiri, Ingrid Moerman, and Jeroen Hoebeke. 2024. Simulating and validating openwifi W-TSN in ns-3. In *IEEE WFCS2024, the 20th IEEE International Conference on Factory Communication Systems*. 1–4.

[94] Maryam Pahlevan, Jonas Schmeck, and Roman Obermaisser. 2019. Evaluation of TSN dynamic configuration model for safety-critical applications. In *2019 IEEE Intl. Conf. on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*. IEEE, 566–571.

[95] Yungang Pan, Rouhollah Mahfouzi, Soheil Samii, Petru Eles, and Zebo Peng. 2024. Multi-traffic resource optimization for real-time applications with 5G configured grant scheduling. *ACM Transactions on Embedded Computing Systems* (2024).

[96] Taeju Park, Soheil Samii, and Kang G. Shin. 2019. Design optimization of frame preemption in real-time switched Ethernet. In *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE'19)*. IEEE, 420–425.

[97] Yifei Peng, Xinlin Wu, Xiaodong Tu, Tigang Jiang, Du Xu, and Jun Xie. 2023. A distributed TSN time synchronization algorithm with increased tolerance for failure scenarios. In *2023 9th International Conference on Computer and Communications (ICCC'23)*. IEEE, 493–499.

[98] Julius Pfrommer, Andreas Ebner, Siddharth Ravikumar, and Bhagath Karunakaran. 2018. Open source OPC UA PubSub over TSN for realtime industrial communication. In *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA'18)*, Vol. 1. IEEE, 1087–1090.

[99] PROFIBUS & PROFINET International (PI) 2021. *PROFINET over TSN Guideline*. PROFIBUS & PROFINET International (PI). White Paper.

[100] Gunnar Prytz. 2008. A performance analysis of EtherCAT and PROFINET IRT. In *2008 IEEE International Conference on Emerging Technologies and Factory Automation*. IEEE, 408–415.

[101] Li Qiang, Xuesong Geng, Bingyang Liu, Toerless Eckert, Liang Geng, and Guangpeng Li. 2019. Large-scale deterministic IP network. *Internet-Draft draft-qiang-detnet-large-scale-detnet-05*. Internet Engineering Task Force. https://datatracker.ietf.org/doc/draft-qiang-detnet-large-scaledetnet/05/. Work in Progress.

[102] Wei Quan, Wenwen Fu, Jinli Yan, and Zhigang Sun. 2020. OpenTSN: An open-source project for time-sensitive networking system development. *CCF Transactions on Networking* 3, 1 (2020), 51–65.

[103] Niklas Reusch, Mohammadreza Barzegaran, Luxi Zhao, Silviu S. Craciunas, and Paul Pop. 2023. Configuration optimization for heterogeneous time-sensitive networks. *Real-Time Systems* 59, 4 (2023), 705–747.

[104] Alexey M. Romanov, Francesco Gringoli, and Axel Sikora. 2021. A precise synchronization method for future wireless TSN networks. *IEEE Transactions on Industrial Informatics* 17, 5 (2021), 3682–3692.

[105] Zenepe Satka, Mohammad Ashjaei, Hossein Fotouhi, Masoud Daneshtalab, Mikael Sjödin, and Saad Mubeen. 2023. A comprehensive systematic review of integration of time sensitive networking and 5G communication. *Journal of Systems Architecture* 138 (2023), 102852.

[106] Christian Schlegel, Anton Meindl, Stefan Schönegger, Bhagath Karunakaran, Huazhen Song, and Stéphane Potier. 2011. *Industrial Ethernet Facts - The 5 Major Technologies. Whitepaper. POWERLINK* 6 (2011).

[107] Sebastian Schriegel, Alexander Biendarra, Thomas Kobzan, Ludwig Leurs, and Jürgen Jasperneite. 2020. Ethernet TSN Nano Profil–Migrationshelfer vom industriellen Brownfield zum Ethernet TSN-basierten IIoT. In *Kommunikation und Bildverarbeitung in der Automation*. Springer, 171–188.

[108] Sebastian Schriegel and Juergen Jasperneite. 2021. A migration strategy for profinet toward Ethernet TSN-based field-level communication: An approach to accelerate the adoption of converged IT/OT communication. *IEEE Industrial Electronics Magazine* 15, 4 (2021), 43–53.

[109] Sebastian Schriegel, Thomas Kobzan, and Jürgen Jasperneite. 2018. Investigation on a distributed SDN control plane architecture for heterogeneous time sensitive networks. In *2018 14th IEEE International Workshop on Factory Communication Systems (WFCS'18)*. IEEE, 1–10.

[110] Oscar Seijo, Xabier Iturbe, and Inaki Val. 2021. Tackling the challenges of the integration of wired and wireless TSN with a technology proof-of-concept. *IEEE Transactions on Industrial Informatics* 18, 10 (2021), 7361–7372.

[111] Óscar Seijo, Jesús Alberto López-Fernández, and Iñaki Val. 2020. w-SHARP: Implementation of a high-performance wireless time-sensitive network for low latency and ultra-low cycle time industrial applications. *IEEE Transactions on Industrial Informatics* 17, 5 (2020), 3651–3662.

[112] Ritu Sethi, Akshay Kadam, Krishnanand Prabhu, and Nikhileswar Kota. 2022. Security considerations to enable time-sensitive networking over 5G. *IEEE Open Journal of Vehicular Technology* 3 (2022), 399–407.

[113] Dawei Shen, Tianyu Zhang, Jiachen Wang, Qingxu Deng, Song Han, and Xiaobo Sharon Hu. 2022. Distributed successive packet scheduling for multi-channel real-time wireless networks. In *2022 IEEE 28th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA'22)*. IEEE, 71–80.

[114] Dawei Shen, Tianyu Zhang, Jiachen Wang, Qingxu Deng, Song Han, and Xiaobo Sharon Hu. 2022. QoS guaranteed resource allocation for coexisting eMBB and URLLC traffic in 5G industrial networks. In *2022 IEEE 28th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA'22)*. IEEE, 81–90.

[115] Boxin Shi, Xiaodong Tu, Bin Wu, and Yifei Peng. 2023. Recent advances in time-sensitive network configuration management: A literature review. *Journal of Sensor and Actuator Networks* 12, 4 (2023), 52.

[116] Haochuan Shi, Adnan Aijaz, and Nan Jiang. 2021. Evaluating the performance of over-the-air time synchronization for 5G and TSN integration. In *2021 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom'21)*. IEEE, 1–6.

[117] Emiliano Sisinni, Abusayeed Saifullah, Song Han, Ulf Jennehag, and Mikael Gidlund. 2018. Industrial internet of things: Challenges, opportunities, and directions. *IEEE Transactions on Industrial Informatics* 14, 11 (2018), 4724–4734.

[118] Johannes Specht and Soheil Samii. 2016. Urgency-based scheduler for time-sensitive switched Ethernet networks. In *2016 28th Euromicro Conference on Real-Time Systems (ECRTS'16)*. IEEE, 75–85.

[119] Kevin B. Stanton. 2018. Distributing deterministic, accurate time for tightly coordinated network and software applications: IEEE 802.1 AS, the TSN profile of PTP. *IEEE Communications Standards Magazine* 2, 2 (2018), 34–40.

[120] Wilfried Steiner, Günther Bauer, Brendan Hall, Michael Paulitsch, and Srivatsan Varadarajan. 2009. TTEthernet dataflow concept. In *2009 Eighth IEEE International Symposium on Network Computing and Applications*. IEEE, 319–322.

[121] Tobias Striffler, Nicola Michailow, and Michael Bahr. 2019. Time-sensitive networking in 5th generation cellular networks-current state and open topics. In *2019 IEEE 2nd 5G World Forum (5GWF'19)*. IEEE, 547–552.

[122] Tobias Striffler and Hans D. Schotten. 2021. The 5G transparent clock: Synchronization errors in integrated 5G-TSN industrial networks. In *2021 IEEE 19th International Conference on Industrial Informatics (INDIN'21)*. IEEE, 1–6.

[123] Susruth Sudhakaran, Karl Montgomery, Mohamed Kashef, Dave Cavalcanti, and Richard Candell. 2021. Wireless time sensitive networking for industrial collaborative robotic workcells. In *2021 17th IEEE International Conference on Factory Communication Systems (WFCS'21)*. IEEE, 91–94.

[124] Susruth Sudhakaran, Karl Montgomery, Mohamed Kashef, Dave Cavalcanti, and Richard Candell. 2022. Wireless time sensitive networking impact on an industrial collaborative robotic workcell. *IEEE Transactions on Industrial Informatics* 18, 10 (2022), 7351–7360.

[125] Texas Instruments 2023. *Time Sensitive Networking for Industrial Automation*. Texas Instruments. White Paper.

[126] Minh-Thuyen Thi, Sosthène Guédon, Siwar Ben Hadj Said, Michael Boc, David Miras, Jean-Baptiste Dore, Marc Laugeois, Xavier Popon, and Benoit Miscopein. 2022. IEEE 802.1 TSN time synchronization over Wi-Fi and 5G mobile networks. In *2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall'22)*. IEEE, 1–7.

[127] Daniel Thiele, Rolf Ernst, and Jonas Diemer. 2015. Formal worst-case timing analysis of Ethernet TSN's time-aware and peristaltic shapers. In *2015 IEEE Vehicular Networking Conference (VNC'15)*. IEEE, 251–258.

[128] J.-P. Thomesse. 2005. Fieldbus technology and industrial automation. In *2005 IEEE Conference on Emerging Technologies and Factory Automation*, Vol. 1. IEEE, 651–653.

[129] Shuo Tian and Yihong Hu. 2019. The role of OPC UA TSN in IT and OT convergence. In *2019 Chinese Automation Congress (CAC'19)*. IEEE, 2272–2276.

[130] Wenbin Tian, Chaojie Gu, Miao Guo, Shibo He, Jiawen Kang, Dusit Niyato, and Jiming Chen. 2024. Large-scale deterministic networks: Architecture, enabling technologies, case study and future directions. *IEEE Network* (2024).

[131] Hristo Trifonov and Donal Heffernan. 2023. OPC UA TSN: A next-generation network for Industry 4.0 and IIoT. *International Journal of Pervasive Computing and Communications* 19, 3 (2023), 386–411.

[132] Marian Ulbricht, Stefan Senk, Hosein K. Nazari, How-Hang Liu, Martin Reisslein, Giang T. Nguyen, and Frank H. P. Fitzek. 2023. TSN-FlexTest: Flexible TSN measurement testbed. *IEEE Transactions on Network and Service Management* (2023).

[133] János Farkas, Nigel Bragg, Paul Unbehagen, Glenn Parsons, Peter J. Ashwood-Smith, and Chris Bowers. 2016. IS-IS path control and reservation. RFC 7813. https://doi.org/10.17487/RFC7813

[134] Inaki Val, Oscar Seijo, Raul Torrego, and Armando Astarloa. 2022. IEEE 802.1 AS clock synchronization performance evaluation of an integrated wired–wireless TSN architecture. *IEEE Transactions on Industrial Informatics* 18, 5 (2022), 2986–2999.

[135] Srinidhi Varadarajan and Tzi-cker Chiueh. 1998. EtheReal: A host-transparent real-time Fast Ethernet switch. In *Proceedings Sixth International Conference on Network Protocols (Cat. No. 98TB100256)*. IEEE, 12–21.

[136] Balázs Varga, János Farkas, Ferenc Fejes, Junaid Ansari, István Moldován, and Miklós Máté. 2023. Robustness and reliability provided by deterministic packet networks (TSN and DetNet). *IEEE Transactions on Network and Service Management* (2023).

[137] Balazs Varga, János Farkas, Andrew G. Malis, and Stewart Bryant. 2021. *Deterministic Networking (DetNet) Data Plane: IP over IEEE 802.1 Time-Sensitive Networking (TSN)*. Technical Report. RFC 9023. https://doi.org/10.17487/RFC9023

[138] Christian von Arnim, Mihai Drăgan, Florian Frick, Armin Lechler, Oliver Riedel, and Alexander Verl. 2020. TSN-based converged industrial networks: Evolutionary steps and migration paths. In *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'20)*, Vol. 1. IEEE, 294–301.

[139] Stefan Waldhauser, Benedikt Jaeger, and Max Helm. 2020. Time synchronization in time-sensitive networking. *Network* 51 (2020), 51–56.

[140] Jiachen Wang, Tianyu Zhang, Xiaobo Sharon Hu, and Song Han. 2023. Resource virtualization with end-to-end timing guarantees for multi-hop multi-channel real-time wireless networks. In *2023 IEEE Real-Time Systems Symposium (RTSS'23)*. IEEE, 385–396.

[141] Jiachen Wang, Tianyu Zhang, Dawei Shen, Xiaobo Sharon Hu, and Song Han. 2021. APaS: An adaptive partition-based scheduling framework for 6TiSCH networks. In *2021 IEEE 27th Real-Time and Embedded Technology and Applications Symposium (RTAS'21)*. IEEE, 320–332.

[142] Jiachen Wang, Tianyu Zhang, Dawei Shen, Xiaobo Sharon Hu, and Song Han. 2022. HARP: Hierarchical resource partitioning in dynamic industrial wireless networks. In *2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS'22)*. IEEE, 1029–1039.

[143] Shouliang Wang, Qimin Xu, Yanzhou Zhang, Lei Xu, and Cailian Chen. 2022. Hybrid traffic scheduling based on adaptive time slot slicing in time-sensitive networking. In *2022 IEEE International Conference on Industrial Technology (ICIT'22)*. IEEE, 1–7.

[144] Yue Wang, Yu-Jing Wu, Yi-Hu Xu, and Yi-Nan Xu. 2023. Research of in-vehicle Ethernet information fault detection based on IEEE 802.1 Qci. *International Journal of Future Computer and Communication* 12, 1 (2023).

[145] Karl Weber. 2018. *EtherCAT and TSN - Best Practices for Industrial Ethernet System Architectures*. EtherCAT Technology Group. White Paper.

[146] Yi-Hung Wei, Quan Leng, Song Han, Aloysius K. Mok, Wenlong Zhang, and Masayoshi Tomizuka. 2013. RT-WiFi: Real-time high-speed communication protocol for wireless cyber-physical control applications. In *2013 IEEE 34th Real-Time Systems Symposium*. IEEE, 140–149.

[147] Dominik Welte, Christopher Lehmann, Manuel Schappacher, Thomas Höschele, Axel Sikora, and Frank H. P. Fitzek. 2024. TSN over 5G: Overcoming challenges and realizing integration. In *2024 IEEE 20th International Conference on Factory Communication Systems (WFCS'24)*. IEEE, 1–8.

[148] Martin Wollschlaeger, Thilo Sauter, and Juergen Jasperneite. 2017. The future of industrial communication: Automation networks in the era of the internet of things and Industry 4.0. *IEEE Industrial Electronics Magazine* 11, 1 (2017), 17–27.

[149] Jordon Woods, Steve Zuponcic, and Rockwell Automation. 2017. QoS-application of TSN to EtherNet/IP networks. *ODVA, Texas, USA* (2017).

[150] Lukas Wüsteney, David Hellmanns, Markus Schramm, Lukas Osswald, René Hummen, Michael Menth, and Tobias Heer. 2022. Analyzing and modeling the latency and jitter behavior of mixed industrial TSN and DetNet networks. In *Proceedings of the 18th International Conference on emerging Networking EXperiments and Technologies*. 91–109.

[151] Yanli Xu, Jian Shang, and Hao Tang. 2023. Recent trends of in-vehicle time sensitive networking technologies, applications and challenges. *China Communications* 20, 11 (2023), 30–55.

[152] Chuanyu Xue. 2024. Open-source Toolkit for TSN Scheduling Algorithm Design and Evaluation. https://github.com/ChuanyuXue/tsnkit

[153] Chuanyu Xue, Tianyu Zhang, and Song Han. 2024. Towards cost-effective real-time high-throughput end station design for time-sensitive networking (TSN). In *2024 61st ACM/IEEE Design Automation Conference (DAC'24)*. IEEE.

[154] Chuanyu Xue, Tianyu Zhang, Yuanbin Zhou, Mark Nixon, Andrew Loveless, and Song Han. 2023. Real-time scheduling for 802.1Qbv Time-Sensitive Networking (TSN): A systematic review and experimental study. *arXiv preprint arXiv:2305.16772* (2023).

[155] Chuanyu Xue, Tianyu Zhang, Yuanbin Zhou, Mark Nixon, Andrew Loveless, and Song Han. 2024. Real-time scheduling for 802.1Qbv Time-Sensitive Networking (TSN): A systematic review and experimental study. In *Proceedings of the 30th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'24)*.

[156] Yiheng Yang, Chenlong Yao, Guochu Shou, Yaqiong Liu, Zhifu Li, Hongxing Li, and Yihong Hu. 2024. Clock drift and time error compensation with neural network for time synchronization in TSN. In *2024 IEEE 7th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC'24)*, Vol. 7. IEEE, 440–444.

[157] Zhilan Yu and Zhi Li. 2009. Best master clock algorithm of precision clock synchronization protocol. *Electric Power Automation Equipment* (2009).

[158] Zelin Yun, Peng Wu, Shengli Zhou, Aloysius K. Mok, Mark Nixon, and Song Han. 2022. RT-WiFi on software-defined radio: Design and implementation. In *2022 IEEE 28th Real-Time and Embedded Technology and Applications Symposium (RTAS'22)*. IEEE, 254–266.

[159] Jiayi Zhang, Lihao Chen, Tongtong Wang, and Xinyuan Wang. 2019. Analysis of TSN for industrial automation based on network calculus. In *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'19)*. IEEE, 240–247.

[160] Jinglong Zhang, Qimin Xu, Xuanzhao Lu, Yajing Zhang, and Cailian Chen. 2020. Coordinated data transmission in time-sensitive networking for mixed time-sensitive applications. In *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 3805–3810.

[161] Litianyi Zhang, Yifan Gu, Rui Wang, Kan Yu, Zhibo Pang, Yonghui Li, and Branka Vucetic. 2022. Enabling real-time quality-of-service and fine-grained aggregation for wireless TSN. *Sensors* 22, 10 (2022), 3901.

[162] Liwei Zhang, Tong Zhang, Wenxue Wu, Xiaoqin Feng, Guoxi Lin, and Fengyuan Ren. 2024. Fault-tolerant cyclic queuing and forwarding in time-sensitive networking. In *2024 Design, Automation & Test in Europe Conference & Exhibition (DATE'24)*. IEEE, 1–6.

[163] Tianyu Zhang, Tao Gong, Song Han, Qingxu Deng, and Xiaobo Sharon Hu. 2018. Distributed dynamic packet scheduling framework for handling disturbances in real-time wireless networks. *IEEE Transactions on Mobile Computing* 18, 11 (2018), 2502–2517.

[164] Tianyu Zhang, Tao Gong, Song Han, Qingxu Deng, and Xiaobo Sharon Hu. 2019. Fully distributed packet scheduling framework for handling disturbances in lossy real-time wireless networks. *IEEE Transactions on Mobile Computing* 20, 2 (2019), 502–518.

[165] Tianyu Zhang, Tao Gong, Xiaobo Sharon Hu, Qingxu Deng, and Song Han. 2021. Dynamic resource management in real-time wireless networks. *Wireless Networks and Industrial IoT: Applications, Challenges and Enablers* (2021), 131–156.

[166] Tianyu Zhang, Tao Gong, Mingsong Lyu, Nan Guan, Song Han, and Xiaobo Sharon Hu. 2022. Reliable dynamic packet scheduling with slot sharing for real-time wireless networks. *IEEE Transactions on Mobile Computing* (2022).

[167] Tianyu Zhang, Xiaobo Sharon Hu, and Song Han. 2023. Contention-free configured grant scheduling for 5G URLLC traffic. In *2023 60th ACM/IEEE Design Automation Conference (DAC'23)*. IEEE, 1–6.

[168] Tianyu Zhang, Jiachen Wang, Xiaobo Sharon Hu, and Song Han. 2023. Real-time flow scheduling in industrial 5G new radio. In *2023 IEEE Real-Time Systems Symposium (RTSS'23)*. IEEE, 371–384.

[169] Tianyu Zhang, Chuanyu Xue, Jiachen Wang, Zelin Yun, Natong Lin, and Song Han. 2024. A survey on industrial Internet of Things (IIoT) testbeds for connectivity research. *arXiv preprint arXiv:2404.17485* (2024).

[170] Lin Zhao, Feng He, Ershuai Li, and Jun Lu. 2018. Comparison of time sensitive networking (TSN) and TTEthernet. In *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC'18)*. IEEE, 1–7.

[171] Luxi Zhao, Paul Pop, and Sebastian Steinhorst. 2022. Quantitative performance comparison of various traffic shapers in time-sensitive networking. *IEEE Transactions on Network and Service Management* 19, 3 (2022), 2899–2928.

[172] Zifan Zhou, Michael Stübert Berger, Sarah Renée Ruepp, and Ying Yan. 2019. Insight into the IEEE 802.1 Qcr asynchronous traffic shaping in time sensitive network. *Advances in Science, Technology and Engineering Systems Journal* 4, 1 (2019), 292–301.

[173] Zifan Zhou, Juho Lee, Michael Stübert Berger, Sungkwon Park, and Ying Yan. 2021. Simulating TSN traffic scheduling and shaping for future automotive Ethernet. *Journal of Communications and Networks* 23, 1 (2021), 53–62.

[174] Zifan Zhou, Ying Yan, Sarah Ruepp, and Michael Berger. 2017. Analysis and implementation of packet preemption for time sensitive networks. In *2017 IEEE 18th International Conference on High Performance Switching and Routing (HPSR'17)*. IEEE, 1–6.

Received 22 August 2023; revised 24 July 2024; accepted 26 August 2024