Advances in Time-Sensitive Networking Research: Postprint

Authors: Cao Zhipeng, Liu Qinrang, LIU Dongpei, Yan Binghao

Date: 2020-09-28T00:00:00+00:00

Abstract

With the development of transmission technologies and the increase in application scenarios, Time-Sensitive Networking (TSN) has attracted widespread attention from various sectors. This paper first introduces the development history and technical content of TSN; then, it presents some important mechanisms in the TSN standard protocols, and focuses on reviewing the research status of TSN both domestically and internationally in recent years, analyzing the application characteristics in different scenarios; finally, it points out the possible future development directions of TSN.

Full Text

Preamble

Survey of Time-Sensitive Networking

Cao Zhipeng, Liu Qinrang, Liu Dongpei, Yan Binghao
(PLA Strategic Support Force Information Engineering University, Zhengzhou, Henan 450003, China)

Abstract: With the development of transmission technology and the increase in application scenarios, Time-Sensitive Networking (TSN) has attracted widespread attention across various sectors. This paper first introduces the development history and technical content of TSN, then elaborates on several important mechanisms in TSN standard protocols, with a focused review of recent research progress on TSN both domestically and internationally, analyzing application characteristics across different scenarios. Finally, it points out potential future development directions for TSN.

Keywords: time-sensitive networking; deterministic networking; ethernet; quality of service

With continuous technological advancement, new information technologies are emerging incessantly, such as 5G, Internet of Things, autonomous driving, next-generation audio/video transmission, and industrial control. In these application scenarios, certain data flows require deterministic transmission, posing significant challenges to traditional transmission technologies [?]. Conventional data transmission technologies suffer from issues including complex equipment, numerous protocols, proprietary technologies, and poor compatibility between different vendors and solution providers, making widespread deployment and equipment upgrades difficult. To meet the demands of various applications for low latency, low jitter, congestion-free, lossless, and high-robustness data transmission, Time-Sensitive Networking (TSN) has emerged as a solution [?].

How to efficiently and accurately handle deterministic data transmission has always been a critical concern in network technology development. As a novel approach to solving deterministic data transmission, TSN has garnered extensive attention from both academia and industry. The core of TSN consists of a series of supplementary protocols proposed by IEEE for 802.1 bridged networks, supporting mixed transmission of best-effort traffic and time-sensitive traffic on the same physical medium—i.e., converged network transmission. The IEEE TSN protocol standards specify fundamental frameworks for mechanisms including time synchronization, transmission scheduling, path control, resource reservation, and reliability.

TSN primarily operates at the data link layer, with its advantage being the establishment of universal standards for deterministic data transmission in converged networks, providing clear direction for new technology development and enhancing compatibility among various Ethernet-based solutions. Currently, TSN products are gradually entering the market, such as testbeds and switch chips supporting TSN standard protocols. This paper introduces relevant work on TSN, summarizes its development background, key mechanisms, research hotspots, and application status, and discusses its future development trends.

1.1 Development Background of Time-Sensitive Networking

Nowadays, the demand for deterministic, low-latency, and high-robustness data transmission is increasing, particularly in control systems, industrial internet, 5G fronthaul, and in-vehicle networks. Some critical traffic flows are highly sensitive to timeliness and require timely and accurate delivery. For instance, in industrial control networks, failure consequences are more severe compared to traditional internet, with higher requirements for stability and determinism. These networks often transmit relatively short data packets with round-trip times between 250 s and 10ms [?]. In 5G transmission, the URLLC (ultra-reliable and low latency) use case has been proposed, which demands extremely high stability, latency, and reliability. For example, data latency in the user plane must not exceed 1ms, latency in the control plane must not exceed 20ms, and the success rate of transmitting 32-bit protocol unit data within 1ms must be no less than 99.999% [?].

To address such data transmission challenges, academia and industry have proposed numerous solutions, evolving from fieldbus to industrial Ethernet, and then to real-time industrial Ethernet. Furthermore, open interconnection, open-source, integration, standardization, and software-defined capabilities are major trends in current technology development. Through standards formulated by relevant organizations, more participants can engage in technical discussions, facilitating rapid technology development and breaking down technical barriers between different solutions.

1.2 Challenges Faced by Traditional Technologies

Current widely-applied solutions for deterministic data transmission include FlexRay [?] in the automotive domain, ARINC664 p7 [?] in avionics, and PROFINET [?], EtherCAT [?], and Powerlink [?] in industrial control—all of which are Ethernet-based real-time control transmission systems.

However, with technological evolution, numerous new technologies such as autonomous driving, industrial automation, and ADAS (advanced driver assistance systems) have emerged, demanding extremely high bandwidth and low latency. For example, autonomous driving devices contain numerous sensors, requiring massive data transmission within extremely short timeframes—the higher the system transmission rate, the better. Generally, image processing requires at least 100Mbps transmission rate, while CAN' s transmission rate does not exceed 1Mbps and FlexRay' s does not exceed 10Mbps [?], making them unable to meet requirements.

Moreover, beyond specific technical details, these solutions share common problems: First, although all are based on fieldbus or Ethernet technologies, various solutions differ at every level from software to hardware, with some being proprietary protected technologies. Second, with continuous development in industrial control, various sensors, processors, memory units, and actuators need to be mutually compatible to achieve intercommunication and collaborative work. Finally, the convergence of IT (information technology) and OT (operational technology) is the future trend, with the goal of establishing a data link between manufacturing execution systems (in industrial fields) and operation management systems (in offices) and integrating them on a unified information platform to achieve synergy in production management, operation decision-making, and manufacturing execution, thereby comprehensively improving work efficiency.

1.3 New Solution—Time-Sensitive Networking

To address these issues, Time-Sensitive Networking emerged. In 2006, the IEEE 802.1 working group established the Audio Video Bridging (AVB) task group, focusing on solving time synchronization, low latency, and bandwidth reservation issues in audio/video transmission. AVB requires data transmission delay not to exceed 2ms within 7 hops [?]. AVB technology research provided foundational support for TSN development—in fact, TSN technology' s predecessor was AVB technology.

However, due to its limited technical scope and relatively single application scenario, AVB could not meet data transmission requirements across different scenarios. IEEE improved upon AVB, expanding its vision to various industrial control systems, and established the TSN working group in 2012. TSN represents a supplement and enhancement to IEEE 802.1 standards, aiming to specify universal deterministic transmission mechanisms.

Table 1 lists the officially released TSN-related standards. TSN application scenarios involve complex large-scale networks with limited bandwidth, multiple nodes, and simultaneous generation and transmission of traffic with diverse QoS requirements. For example, the IEC/IEEE 60802 TSN industrial automation profile specifies that for synchronous traffic, the maximum packet length is 100 bytes with end-to-end delay not exceeding 2ms; for cyclic traffic, the maximum packet length is 1000 bytes with delay between 2 and 20ms [?]. The latest standard IEEE Std 802.1Qcp-2018 [?] was released by IEEE in 2018, primarily adding support for YANG data models. TSN standardization work continues, with Table 2 listing standards currently under revision.

Table 1: TSN Related Standards

Standard	Description
IEEE Std 802.1Qav-2009 [?]	Enhanced forwarding and queuing for time-sensitive streams
IEEE Std 802.1Qat-2010 [?]	Stream Reservation Protocol
IEEE Std 802.1Qca-2015 [?]	Path control and reservation
IEEE Std 802.1Qbv-2015 [?]	Enhancements for scheduled traffic
IEEE Std 802.1Qbu-2016 [?]	Frame preemption
IEEE Std 802.1Qch-2017 [?]	Cyclic queuing and forwarding
IEEE Std 802.1Qci-2017 [?]	Per-stream filtering and policing
IEEE Std 802.1Qcc-2018 [?]	Stream reservation protocol enhancements
IEEE Std 802.1Qcp-2018 [?]	YANG data model
IEEE Std 802.1BA-2011 [?]	AVB systems
IEEE Std 802.1AS-2011 [?]	Timing and synchronization for time-sensitive applications
IEEE Std 802.1CB-2017 [?]	Frame replication and elimination for reliability
IEEE Std 802.1CM-2018 [?]	TSN for fronthaul

Table 2: On-going TSN Projects

Project	Description
P802.1Qcj	Automatic attachment to Provider Backbone Bridging (PBB) services
P802.1Qcr	Asynchronous traffic shaping
P802.1Qcw	YANG data model for scheduled traffic, frame preemption, per-stream filtering and policing
P802.1Qcx	YANG data model for connection fault management
P802.1Qcz	Resource allocation protocol
P802.1Qdd	Enhancements to 802.1AB standard
P802.1Qdj	YANG data model for LLDP
P802.1ABcu	Support for multi-frame protocol data units
P802.1ABdh	Supplement/enhancement to 802.1CB standard
P802.1CBcv	YANG data model for per-stream filtering and policing
P802.1CBdb	Extended stream authentication for per-stream filtering and policing
P802.1CMde	Enhancements to fronthaul profiles
IEC/IEEE 60802	TSN profile for industrial automation
P802.1CS	Link-local registration protocol
P802.1CQ	Multicast and local address allocation
P802.1DC	QoS provision for network systems
P802.1DF	TSN profile for service providers
P802.1DG	TSN profile for in-vehicle Ethernet communication
P802.1AS-Rev	Link aggregation revision

IEEE 802.1AS [?] represents further development and application of the IEEE 1588 protocol [?] at the data link layer. The clock synchronization domain is called the gPTP (generalized precision time protocol) domain, where all devices are time-sensitive, including bridges and end stations, and can support multiple heterogeneous networks. In the gPTP domain, any device containing a clock source can be selected as the grand master (GM) through the BMCA (best master clock algorithm). The elected grand master transmits time synchronization-related information to other nodes, which use this information to adjust their time to achieve synchronization. The 802.1AS time synchronization mechanism primarily consists of three parts: path delay measurement, BMCA algorithm, and synchronization time adjustment.

2.1.1 Principle of Path Delay Measurement

Path delay measurement aims to measure the delay between adjacent nodes. In transmission delay measurement, one party is the initiator and the other is the responder. As shown in Figure 1 [Figure 1: see original paper], measurement is achieved through the exchange of PdelayReq, PdelayResp, PdelayRespFollowUp messages and timestamps. The transmission delay can be calculated using Equation (1).

$$d = \frac{(t_4 - t_1) - (t_3 - t_2)}{2}$$

2.1.2 BMCA Algorithm

The BMCA algorithm is used to determine the grand master in the gPTP domain. Clock information for synchronization is transmitted to every node in the system through a spanning tree structure. The election of the best master clock is implemented through the propagation of Announce messages in the system. Announce messages contain status information of each node. Initially, the roles of nodes and their ports are undetermined, but the algorithm can select the best master clock within a finite number of steps and determine the status of each node port [**?**].

The BMCA algorithm consists of numerous state machines that coordinate with each other to achieve best master clock selection and port role confirmation through the transmission of various internal vectors, as shown in Figure 2 [Figure 2: see original paper].

2.1.3 Transmission of Clock Synchronization Information

In the gPTP domain, all devices must synchronize with the master clock. Therefore, each clock needs to transmit its synchronization information and adjust its parameters based on information from other clocks. For example, assuming three nodes A, B, and C, time synchronization information needs to be transmitted from A through B to C, as shown in Figure 3 [Figure 3: see original paper]. The system continuously updates the correctionField and rateRatio through the transmission of Sync and FollowUp messages.

Where preciseOriginTimestamp is the time when the master clock initiates synchronization information, which remains unchanged during information transmission. rateRatio is used to correct differences in clock frequencies, calculated using Equation (2):

$$rateRatio = \frac{frequency_B}{frequency_A}$$

correctionField is used to correct transmission delay and node processing delay during transmission. Nodes farther from the master clock have larger correction values, while closer nodes have smaller values, calculated using Equation (3). As information gradually propagates throughout the entire synchronization time spanning tree, each node achieves synchronization with the master clock.

2.2 Transmission Scheduling Mechanism

The transmission scheduling mechanism is one of TSN' s core technologies, enabling converged network transmission while guaranteeing performance requirements. Key mechanisms include CBS (credit-based shaper), TAS (time-aware shaper), and frame preemption.

2.2.1 CBS Mechanism

IEEE 802.1Qav [?] provides performance guarantees for time-sensitive, loss-sensitive real-time audio/video applications. This standard specifies priority regeneration algorithms and controlled bandwidth queue depletion algorithms, but most importantly defines the CBS mechanism for traffic shaping to prevent high-priority traffic from occupying bandwidth for extended periods and interfering with other streams. The CBS algorithm is queue-specific, providing a method for shaping queue output traffic based on "credit" values. For any queue using the CBS mechanism, data can only be output when its credit value is greater than or equal to 0; otherwise, no output occurs.

When data is being transmitted (i.e., when credit value ≥ 0), the queue' s credit value decreases at the sendSlope rate. When data is not being transmitted (i.e., when credit value < 0, or when higher-priority queues are transmitting), the credit value increases at the idleSlope rate. If credit is positive and no data is being transmitted, credit is immediately set to zero. Both sendSlope and idleSlope are measured in bits per second. The credit value is bounded, ranging between hiCredit (maximum value) and loCredit (minimum value), as shown in Figure 4 [Figure 4: see original paper].

2.2.2 TAS Mechanism

In the IEEE 802.1Qbv [?] standard, TSN provides a TDM-like converged transmission solution called Time-Aware Shaper (TAS). TAS operates based on time synchronization from 802.1AS. According to the protocol, up to 8 different priority queues can be used for transmission, with each queue' s scheduling operating independently. 802.1Qbv adds a special gating mechanism where each queue corresponds to a transmission gate. The gate status determines whether a queue can transmit data frames—data can be transmitted when the gate is open, but not when closed. All transmission gates combined form the Gate Control List (GCL), as shown in Figure 5 [Figure 5: see original paper] (status 0 represents closed, status 1 represents open).

Each output port has its own GCL to control output queues within that port. The transmission gate has final authority over transmission—even if a data frame is selected by certain transmission selection algorithms and has sufficient transmission time, it cannot be transmitted if the gate is closed. To prevent interference between data during status transitions, a "guard band" exists between state switches, during which, except for queues that have already started transmission, all other queues are prohibited from transmitting. Time-sensitive data is only transmitted when the port is idle, ensuring transmission quality. The TAS mechanism is one of TSN' s most core functions, enabling mixed transmission of different traffic types and deterministic transmission of time-sensitive traffic.

2.2.3 Frame Preemption Mechanism

Frame preemption during transmission is specified in IEEE 802.1Qbu [?] and IEEE 802.3br [?]. Frame preemption is primarily used for emergency frame transmission in networks and can only be used when both receiving and transmitting devices support it. The system uses the LLDP (link layer discovery protocol) and Ethertype values to determine whether the current link supports frame preemption.

The frame preemption mechanism allows higher-priority data to interrupt lower-priority data transmission once or multiple times. After high-priority data transmission completes, lower-priority data resumes transmission. Key aspects include preempting current frames and preventing preempted frames from continuing transmission [?]. In the 802.3br protocol, frame preemption occurs in the MAC aggregation sublayer, which divides the MAC layer into eMAC (express MAC) and pMAC (preemptable MAC). eMAC handles time-sensitive traffic transmission, while pMAC handles ordinary traffic transmission, with eMAC having higher priority than pMAC.

When a preemptable frame is being transmitted, if no preemption occurs, it passes directly through pMAC. If preemption occurs, the data to be transmitted is cached in the MAC aggregation sublayer. For multiple preemptions, each frame segment has FCS (frame check sequence) verification. The frame structure of each preemptable frame [?] is shown in Figure 6 [Figure 6: see original paper]. Frame preemption can be applied independently or in combination with other mechanisms, often jointly used with CBS and TAS.

2.3 Path Control Mechanism

OSPF (open shortest path first) is a routing protocol designed for TCP/IP, while IS-IS (intermediate system-to-intermediate system) is a routing protocol designed for OSI (non-TCP/IP) architectures. IS-IS is commonly used in large-scale networks (e.g., carriers and banks) and transmits packets at the data link layer. Compared to OSPF, IS-IS is simpler and more stable, resulting in higher processing efficiency under massive data information conditions [?].

Traditional IS-IS uses shortest-path transmission, whereas IEEE 802.1Qca [?] allows establishing explicit trees that can accurately route traffic to predetermined paths, increasing network resilience and reducing congestion. Compared to SPB (shortest path bridging), 802.1Qca offers more functionality, providing explicit path control, bandwidth reservation, and redundancy improvements for data flows. The protocol details bridging methods for unicast and multicast frames on explicit paths and specifies protocols for determining multiple dynamic topologies.

The protocol provides explicit path control, enabling non-shortest-path usage, and integrates tools for bandwidth reservation and stream reservation on forwarding paths. 802.1Qca provides redundancy control mechanisms, supporting IS-IS to carry control information for synchronization and scheduling functions. It evolved from RSTP (rapid spanning tree protocol), MSTP (multiple spanning tree protocol), and SPB, offering multiple dynamic topologies and specifying explicit paths for data forwarding. The protocol includes built-in mechanisms such as SP (shortest path), ECT (equal cost tree), IST (internal spanning tree), MSTI (multiple spanning tree instance), and ET (explicit tree).

2.4 Resource Reservation Mechanism

The Stream Reservation Protocol (SRP) IEEE 802.1Qat [?] specifies an admission control framework for accepting or rejecting streams based on resource reservation requirements and available network resources. It also defines frameworks for reserving network resources and broadcast streams in packet-switched networks with full-duplex Ethernet links. SRP meets real-time transmission requirements by reserving transmission resources on communication links. The protocol determines required bandwidth based on different stream types. For AVB-compliant switches, 75% of bandwidth resources are allocated to AVB data transmission, with the remaining 25% for best-effort traffic. SRP utilizes a combination of three protocols: MMRP (multiple MAC registration protocol), MVRP (multiple VLAN registration protocol), and MSRP (multiple stream registration protocol), all based on the underlying MRP (multiple registration protocol).

IEEE 802.1Qcc [?] enhances SRP by improving mechanisms to support more streams, enabling configurable SR (stream reservation) classes and streams. Additionally, Qcc provides better stream characterization, supports Layer-3 streams, features deterministic stream reservation convergence, and includes a User Network Interface (UNI) for routing and reservation.

2.5 Reliable Transmission Mechanism

The FRER (frame replication and elimination for reliability) specified in IEEE 802.1CB [?] aims to increase the delivery probability of specific packets. When applied on fixed-topology and congestion-protected paths, it effectively reduces packet loss caused by equipment failures. The standard' s transmission modes include unicast from one initiator to one receiver and multicast from one initiator to multiple receivers. IEEE 802.1CB improves reliability and reduces packet loss by replicating data packets of each stream with sequence numbers, sending multiple copies from the source, and finally reassembling and eliminating duplicates at one or more nodes, as shown in Figure 7 [Figure 7: see original paper].

IEEE 802.1CB deployment is flexible, applicable at both end stations and intermediate nodes. It achieves redundancy through packet replication and elimination, works for both unicast and multicast, and supports intermittent and bulk stream transmission. FRER does not address transmission path creation, stream reservation, or in-order packet delivery—these functions are implemented by other protocols.

3 Research Status of Time-Sensitive Networking

As technology continues to develop, TSN has gained widespread academic attention as a new method for solving deterministic network data transmission. Recent research hotspots focus on scheduling analysis and implementation, network delay analysis, redundancy and fault-tolerant networks, fronthaul networks, and time-sensitive software-defined networking. This research includes exploration of new algorithms, improvement of existing mechanisms, and simulation technology studies.

3.1 Research on Scheduling and Routing

Traditional Ethernet transmission mechanisms lack determinism because frames in switch output port queues must wait due to same-priority and higher-priority frames during transmission, causing delays and uncertainty. IEEE 802.1Qbv defines a time-triggered communication mode TAS where output port queues transmit frames strictly according to GCL timing. A key challenge in implementing TSN is traffic scheduling and routing.

a) Traffic Scheduling Computation: For the TAS mechanism in IEEE 802.1Qbv, the standard only specifies the framework without providing concrete implementation methods. Craciunas et al. [**?**] proposed an implementation method for IEEE 802.1Qbv scheduling, summarizing network parameters into device capability parameters and queue configuration parameters, representing network topology with directed graphs. For time-sensitive traffic, they added TAS mechanism constraints on top of deterministic Ethernet transmission constraints. The core of scheduling was summarized as frame offset calculation and transmission queue allocation. Constraints were divided into basic deterministic network transmission constraints and IEEE 802.1Qbv-specific constraints, converting the problem into solving for the minimum number of queues required to transform an unschedulable system into a schedulable one, using SMT (satisfiability modulo theories) methods.

Oliver et al. [**?**] proposed a solution for GCL computation in IEEE 802.1Qbv, analyzing the suitability of using first-order array theory to formalize system constraints, discussing scheduling optimization problems, and employing SMT/OMT-based solvers for scheduling computation. Results showed that first-order array theory is appropriate, particularly effective for small and medium-sized networks.

b) Traffic Routing Computation: Using CSPF (constrained shortest path first) as a routing method, similar to traditional shortest path algorithms, may cause congestion affecting time-sensitive stream transmission. Mubarak et al. [**?**]

proposed novel routing algorithms LB-DRR and CR-DRR. LB-DRR achieves load balancing across different links and can create multiple transmission paths for replicated streams; CR-DRR can recalculate transmission paths for streams when link congestion occurs. Both improve link load compared to traditional shortest-path and load-balancing routing.

c) Joint Routing and Scheduling Computation: GCL computation is an NP-complete problem. Pahlevan et al. [?] argued that existing solutions ignore the interaction between scheduling and routing, are computationally slow, not scalable, and fail to consider multicast streams, internal stream correlations, and distributed systems. They proposed a genetic algorithm-based joint routing-scheduling algorithm, comparing it with schedule-based methods. This approach simultaneously considers routing and scheduling constraints, reduces guard band quantity, and improves scheduling capability and transmission efficiency for time-triggered traffic.

A key advantage of TAS is enabling converged network transmission—simultaneously transmitting time-sensitive and ordinary data traffic. To achieve this, different data traffic types require temporal or spatial separation. Falk et al. [?] argued that "customized" approaches lead to low correctness and high complexity, while ILP (integer linear programming) methods offer mature approaches with fast computation. They implemented a joint routing-scheduling algorithm using ILP and explored problem scalability. Experiments showed the method' s runtime is sensitive to stream quantity, transmission cycle occupancy, and network topology.

d) Stability-Aware Joint Routing and Scheduling: Ethernet timing characteristics heavily depend on traffic scheduling and routing. Mahfouzi et al. [?] argued that stability considerations (maximum tolerable delay and jitter) should be incorporated during routing and scheduling computation. They used SMT to model the joint routing-scheduling problem, considering stability under real-time transmission, control data transmission, and worst-case scenarios. Based on routing subsets and time-slicing concepts, they employed heuristic algorithms to solve the problem, improving overall computational efficiency.

e) AVB-Aware Joint Routing and Scheduling: Gavrilut et al. [?] addressed the common issue of routing and scheduling algorithms considering only TT traffic while ignoring AVB traffic. They proposed an AVB-aware scheduling algorithm that reduces worst-case delay for AVB traffic while ensuring schedulability of time-triggered traffic. The algorithm combines KSP (K-shortest paths) and GRASP (greedy randomized adaptive search procedures). Through three experimental groups evaluating TT traffic schedulability, GCL computation under routing impact, and joint routing-scheduling with AVB traffic consideration, results demonstrated that only by considering AVB traffic can schedulability results for both TT and AVB traffic be obtained.

3.2 Research on Network Delay Analysis

A core characteristic of TSN is low-latency data transmission. Network delay analysis involves calculating delays under various conditions to determine upper or lower bounds for system performance evaluation or verification.

a) Worst-Case Delay Analysis: TSN traffic can be categorized by importance as TT traffic, AVB traffic, and BE traffic. TT traffic is scheduled according to GCL with the strictest delay requirements, followed by AVB traffic, while BE traffic has minimal requirements. Zhao et al. [**?**] analyzed worst-case delay for AVB traffic in networks, discussing both preemptive and non-preemptive modes while considering TT traffic impact on AVB traffic. Using network calculus theory, they derived service curves for AVB Class A and Class B traffic and proved that AVB credit values in CBS mechanism do not overflow. This method provides more accurate worst-case delay estimates than existing approaches.

Zhang et al. [**?**] used network calculus to analyze upper bounds for time-sensitive traffic in industrial automation scenarios. Different traffic types are shaped through CBS and strict priority mechanisms. By calculating worst-case delay upper bounds, they demonstrated that common industrial automation latency requirements can be met through reasonable parameter configuration.

Determining end-to-end delay upper bounds is crucial. Benammar et al. [**?**] argued that traditional analysis methods such as CPA (compositional performance analysis), FA (forward end-to-end delay analysis), and TA (trajectory approach) produce overly large end-to-end delay estimates. They proposed a solution combining FA with CBS, dividing E2E delay into fixed and variable components. Analysis of linear and clustered topologies effectively reduced end-to-end delay estimation ranges.

b) Best-Case Delay Analysis: Rivera-Verduzco et al. [**?**] argued that in TSN analysis, both worst-case and best-case response times are important, though best-case response time has been less studied. Their method considers that bursty traffic using CBS scheduling may be scheduled earlier when transmission conflicts occur, providing calculation methods for conflict-free best-case frame transmission times.

c) Delay Analysis with CDT Traffic: In practical scenarios, TSN networks simultaneously transmit CDT (control data traffic), AVB data, and BE data, where CDT data has strict timing requirements similar to TT data. Maxim et al. [**?**] noted that while AVB traffic delay analysis is relatively mature, research on AVB delay analysis containing CDT is limited. They established network models based on single and multiple guard windows and traffic models, conducting local analysis of AVB frames. Results showed that burst transmission is limited by CBS mechanisms, and worst-case transmission delay is bounded within a certain range.

d) TSN System Feasibility Analysis: As TSN systems mature, feasibility analysis becomes increasingly important. Zhang et al. [**?**] proposed a TSN feasibility analysis method based on real-time calculus, decomposing TSN systems into data models, resource models, and component models. The method considers delay bounds for critical traffic, node storage bounds, and transmission resource utilization, using frame preemption to resolve conflicts between different priorities.

3.3 Research on Redundancy and Fault-Tolerant Networks

To ensure data transmission success rates and link stability, redundant transmission mechanisms must be designed. TSN proposes the IEEE 802.1CB mechanism, specifying the fundamental framework for seamless redundant frame transmission.

a) Analysis and Simulation of Seamless Redundant Transmission: Tan et al. [?] implemented redundant path transmission for data traffic in different network environments using Core4INET modules in OMNeT++ 5.0, analyzing redundancy' s impact on delay. Experiments showed that implementing redundancy increases data transmission delay to some extent, and frames with longer transmission paths always experience a one-transmission-cycle delay.

Qian et al. [?] conducted detailed analysis of IEEE 802.1CB in network topology, redundancy marking, sequence generation functions, sequence recovery functions, timing mechanisms, and potential error detection. Through functional testing at multiple test points using Visual Studio and CppUTest for simulation, experiments demonstrated that applying 802.1CB to specific network topologies reduces packet loss rates without relying on higher-layer transmission protocols. This transparent characteristic enhances network seamless redundancy.

Simulation provides a low-cost, high-efficiency method for protocol verification, yet mature simulation models for implementing TSN timing and non-timing characteristics are lacking. Based on the Riverbed simulation framework, Pahlevan et al. [?] proposed a simulation model for timing characteristics and redundancy management, incorporating error injection mechanisms to evaluate system stability. Experiments showed that systems using IEEE 802.1CB can eliminate transient link faults and provide bounded end-to-end delay and zero packet loss under permanent link failure conditions.

b) Optimization of Redundant Transmission: The redundant transmission framework proposed in IEEE 802.1CB allows implementation at any switching or end node in the transmission network. Smirnov et al. [?] argued this creates excessive implementation space, requiring trade-offs between redundancy, network load, transmission timing, and hardware costs. They proposed two SAT-decoding-based routing algorithms for generating feasible redundant message routes, extending to multi-objective routing optimization. Experimental comparisons demonstrated significant advantages, enabling automatic system stability optimization.

c) Fault-Tolerant Topology and Routing Computation: For safety-critical real-time systems, Gavrilut et al. [?] used UBS (urgency-based scheduler) traffic types to establish a fault-tolerant topology with redundant links to meet routing overhead reduction, fault tolerance, and timing requirements. They compared heuristic algorithms, GRASP, and constraint programming-based models, finding all significantly reduced overhead. GRASP performed particularly well, producing results within reasonable runtime with good scalability.

3.4 Research on Fronthaul Networks

Fronthaul networks have the highest bandwidth requirements and most stringent latency demands in 5G transport networks. CPRI (common public radio interface) used in fronthaul is a standardized protocol [?] defining digital interfaces between radio equipment control (REC) and radio equipment (RE) in wireless infrastructure base stations, enabling interoperability between different vendor equipment and reducing software investment for wireless service providers. Traditional CPRI technology suffers from high cost, large bandwidth consumption, and static configuration, making Ethernet-based solutions (CPRI on Ethernet, CoE) an attractive alternative.

a) Jitter and Delay Analysis for CoE: Chitimalla et al. [?] encapsulated CPRI data in Ethernet frames, analyzing jitter under different conditions and proposing jitter elimination algorithms achieving microsecond-level jitter. They implemented the algorithm in Verilog on FPGA. For multi-hop transmission within several kilometers in fronthaul networks, CoE encapsulation introduces acceptable minimal delay. Experimental results showed encapsulating fixed-size CoE frames requires tens of microseconds, and redundant transmission can reduce jitter by up to 1 s.

b) M2M Delay Analysis in Fronthaul Networks: 5G networks require converged network transmission—simultaneously carrying fronthaul data and M2M (machine-to-machine) data—which can be implemented using TAS mechanisms. However, TAS exhibits low bandwidth utilization and high delay for low-priority traffic, as GCL continues running even without high-priority traffic. Hisano et al. [?] proposed an improved TAS algorithm called GS-TAS, attaching a GS-frame after fronthaul streams. When a bridge receives a GS-frame, high-priority gate status changes from open to closed, and low-priority status changes conversely. Experimental results showed GS-TAS enables converged data transmission, reducing M2M data delay by 50% while maintaining fronthaul data delay.

c) Comparative Analysis of IHON and TSN: 5G fronthaul has strict requirements for delay and jitter, where any bridging may introduce additional delay that must be eliminated or reduced. Bjornstad et al. [?] argued that for various jitter elimination and delay reduction methods, complexity and applicability are most important—excessive complexity or poor applicability hinder practical implementation. Through detailed analysis of transmission mechanisms in IHON (integrated hybrid optical networks) and TSN, they concluded that IHON has the lowest jitter and minimal complexity, while TSN has lower delay.

3.5 Research on Time-Sensitive Software-Defined Networking

Software-Defined Networking (SDN) is a current research hotspot, fostering new network architecture models. Its core concept involves hardware standardization and separation of control and forwarding planes to enhance network programmability through device abstraction. TSN can combine with SDN' s centralized control and management to realize Time-Sensitive Software-Defined Networking (TSSDN).

a) Integration of SDN and TSN: Nayak et al. [**?**] utilized the TSSDN concept to implement time-triggered and non-time-triggered traffic transmission on the same SDN network, using a logically centralized controller to compute global scheduling problems and ILP methods to solve routing and scheduling problems, enabling end devices to execute user-plane packet processing frameworks with high precision. Simulation revealed that under specific topologies, TSSDN provides deterministic transmission delay with end-to-end delay within 14 s and jitter within 7 s.

Hackel et al. [**?**] argued that TSN networks can leverage SDN' s centralized control advantages to achieve network elasticity, security, and adaptability. They combined TSN' s real-time communication with SDN control plane programmability, mapping and registering TSN streams in SDN controllers and implementing them through OpenFlow. Analysis showed that with appropriate protocol mapping, SDN control overhead does not introduce additional delay, maintaining real-time performance while leveraging SDN advantages.

To adapt and maintain TSN networks, technicians require clear understanding of network devices and data flows, along with configuration and simulation tools, making related work complex. Ben H. S. et al. [**?**] utilized SDN's centralized control characteristics to instantiate the control model proposed in IEEE 802.1Qcc, discussing how to use SDN to accelerate network configuration processes using IEEE 802.1AS configuration as an example.

To reduce queuing delay for traffic in TSSDN, Haur et al. [**?**] proposed a time-sensitive traffic scheduling algorithm that effectively reduces time complexity while guaranteeing scheduling quality. Compared with scheduling under unconstrained routing, shortest-path routing, and fixed-path routing, the method demonstrated significant performance improvements.

b) TSSDN Performance Evaluation: SDN' s centralized control facilitates admission control and network reconfiguration in TSN scenarios, but time overhead must be evaluated due to high delay requirements. Thiele et al. [**?**] modeled and analyzed the CPA (compositional performance analysis) framework for time-sensitive Ethernet, experimentally evaluating different protocols. Tests showed that in worst-case scenarios, system transmission delay does not exceed 50ms, demonstrating SDN' s viability for Ethernet admission control and network reconfiguration scenarios.

c) SDN-Based TSN Delay Measurement: For the TAS mechanism in IEEE 802.1Qbv, obtaining delay between adjacent nodes is crucial. Jia et al. [?] defined Time-Sensitive Network Delay (TSND), quantifying the impact of clock synchronization and packet size on TSND measurement, and proposed an SDN-based TSN delay measurement scheme. Experiments showed TSND can effectively predict reception time slots, with maximum measurement error controlled within two time slots. For streams with large packet length variations, using time slots calculated from large delays yields smaller jitter. The proposed TSND measurement method can effectively establish delay databases between adjacent nodes.

3.6 Research on TSN Switches

To ensure network throughput, delay, jitter, and packet loss rate, all nodes in the network must provide deterministic services for time-sensitive traffic. However, implementing TSN standards on high-port-density and high-interface-speed devices is relatively difficult. Pruski et al. [?] proposed an implementation structure for ASIC or FPGA, designing TSN-capable switching nodes by discussing buffer structures and memory bandwidth requirements. The design supports frame preemption, CBS, and TAS mechanisms while effectively reducing hardware overhead.

Due to TSN' s high determinism, reliability, and bandwidth, it can serve as the core network for industrial IoT. However, such core networks must process large volumes of real-time data, and current mature TSN switches can only deterministically schedule up to 1024 real-time streams. Jin et al. [?] addressed this limitation by proposing a method supporting more real-time streams. They used SMT to formulate packet transmission processes and OMT for optimization, designing heuristic algorithms for scheduling conflicts. Compared with existing algorithms, this method requires only 1/20 of the schedule entries for scheduling the same streams.

4 Application Status of Time-Sensitive Networking

4.1 Automotive Domain

With continuous progress in information technology, advanced concepts such as intelligent transportation, vehicle networking, and autonomous driving have gradually emerged. In-vehicle electronic systems are becoming increasingly important. For example, the widely-discussed ADAS system requires massive data transmission and processing [?]. ADAS systems analyze driving environments inside and outside the vehicle through various sensors installed on the vehicle, collecting and analyzing data during operation to implement emergency braking, prevent accidental collisions and lane departure, and enable assisted parking, adaptive cruise control, and driver fatigue detection. These functions impose strict requirements on in-vehicle network data transmission quality.

Traditional in-vehicle network technologies include CAN bus, LIN, FlexRay, and MOST [?]. CAN bus is primarily used for braking, engine, and suspension systems; LIN for lighting, doors, and remote control; FlexRay for engine control, ABS, and suspension; and MOST as the preferred protocol for in-vehicle multimedia. With the development of automotive electronics, these networks' shortcomings—incompatibility, limited bandwidth, high latency, and jitter—have become increasingly prominent. As an Ethernet-based interconnection technology, TSN offers high flexibility and scalability, gradually becoming a new solution. Using TSN as a universal in-vehicle network can simplify different in-vehicle network architectures while guaranteeing performance, latency, and bandwidth, ensuring that control information, audio/video information, and entertainment information all receive good quality of service.

Currently, TSN-enabled in-vehicle devices are gradually entering the market. For example, Marvell' s 88Q5050 automotive Ethernet switch chip is an 8-port, high-security automotive gigabit Ethernet switch chip fully compliant with IEEE 802.3 and 802.1 standards, supporting IEEE 802.1Qav and Qbv functions in TSN protocols and enabling supervision and rate limiting of AVB streams at input ports. Broadcom has also launched the BCM5316x series switch chips supporting IEEE 802.1AS, 802.1Qbv, 802.1Qav, 802.1Qci, and other protocols, primarily targeting unmanned vehicles, drones, robots, and Level 3 autonomous driving.

4.2 Industrial Control Domain

Industry is paramount to national development. The United States proposed the Advanced Manufacturing Partnership (AMP) program in 2011, Germany introduced the Industry 4.0 concept at the 2013 Hannover Messe, and China launched the Made in China 2025 initiative in 2015 [?]. These industrial development plans envision smarter, more advanced future manufacturing with improvements in production systems, logistics management, human-machine interaction, and quality monitoring. With network technology development, we have entered the information age, and integrating industry with internet technology is the general trend.

Many industrial control applications require high real-time performance, determinism, and robustness. Ethernet-based TSN technology not only meets these requirements but also provides solutions for connecting industrial Ethernet to data centers. TSN enables efficient real-time decision-making information and, compared to existing industrial Ethernet technologies, is an open standard that facilitates equipment upgrades, reduces costs, and achieves better compatibility. Traditional control technologies are proprietary to various manufacturers and equipment providers, with different control systems potentially incompatible and difficult to upgrade. TSN addresses these issues in industrial control transmission protocols.

Currently, many renowned IT and automation vendors have begun adopting TSN technology, establishing numerous cooperation organizations. At the 2018 Hannover Messe in Germany, over 20 well-known organizations and vendors including the Avnu Alliance, Alliance Industrial Internet (AII), Edge Computing Consortium (ECC), Huawei, Schneider Electric, National Instruments, B&R, and TTTech released a TSN+OPC UA smart manufacturing testbed. The testbed integrates TSN technology and OPC UA standards, enabling scenarios including predictive maintenance networks, motor synchronization, plotting motion control, and OPC UA over TSN. TSN facilitates IT and OT convergence, improving industrial equipment connectivity and universality, creating broad pathways for further industrial development.

4.3 Telecommunications Domain

With smartphone popularization and development, mobile data demand is increasing with higher requirements. 4G communication cannot meet user needs in many scenarios, making 5G networks the focus. 5G networks offer high transmission rates for massive data transmission applications, featuring low latency, large network capacity, high traffic density, and high connection density. Fifth-generation mobile communication has become a key focus for industry and academia. 2019 was called the "first year of 5G," and in June 2019, China's Ministry of Industry and Information Technology officially issued 5G commercial licenses to China Unicom, China Mobile, China Telecom, and China Broadcasting Network, marking the critical moment for 5G commercial deployment.

At the end of 2018, IEEE released the TSN standard for fronthaul networks—IEEE 802.1CM [?]—primarily addressing fronthaul network construction using Ethernet. Main application scenarios include cloud-based radio access networks and 5G small cell networks, connecting cellular radio equipment with remote controllers. Traditional fronthaul solutions use CPRI interfaces for equipment connection, but CPRI deployment is costly, bandwidth-intensive, and only statically configurable. Using Ethernet-based solutions to build fronthaul networks reduces costs and facilitates configuration, making Ethernet-based CPRI a new solution. The CoE solution, combined with TSN technologies, can meet CPRI interface's strict delay and jitter requirements.

5 Development Trends of Time-Sensitive Networking

As a highly promising technology, TSN uses combinations of multiple standard protocols to address traditional Ethernet's deficiencies in deterministic transmission, attracting widespread attention from academia and industry. However, TSN technologies are still developing, with new standards being drafted and discussed. This paper analyzes and summarizes currently released TSN standard protocols and introduces industry application status. While existing TSN technologies have solved time-sensitive stream transmission problems to some extent, many issues remain before full deployment. TSN research should focus on the following aspects:

1) **Engineering Application and Implementation:** Some TSN mechanisms only provide technical framework standards without detailed implementation specifications. How to configure and implement these standards is a direct research direction. This requires analyzing application scenarios to clarify required mechanisms for specific problems and exploring new methods for configuration scalability research. As an emerging technology just entering the market, extensive work is needed from theory to practice, including implementation methods, performance analysis, simulation technology, and product prototyping.

2) **Enhanced Interconnectivity:** With technological development, computing, communication, and control technologies will converge in CPS (cyber-physical systems). Currently, TSN is often applied in relatively closed LAN scenarios lacking methods to connect closed TSN networks with external networks. The software-defined interconnection technology proposed by the National Digital Switching System Engineering Research Center [**?**] enables software-definable architecture and processing nodes, transforming rigid devices into flexible ones under high energy efficiency. Supporting software-definable interconnection protocols, port types, topology structures, bandwidth allocation, and interconnection modes, it can serve as a solution for future TSN interconnection with homogeneous or heterogeneous networks.

3) **Enhanced Security and Reliability:** TSN application scenarios demand extremely high system transmission security and stability. The IEEE 802.1 TSN working group and 802.1 security working group have begun related research to enhance TSN security. As TSN interconnectivity improves, open networks expose TSN systems to common attacks. If a node on the transmission path fails, end-to-end flow control characteristics may become impossible. Therefore, TSN networks require research on threat detection and node migration technologies. The cyberspace mimic defense concept proposed by Wu Jiangxing et al. [**?**] transforms non-traditional security threats into classic reliability and robustness problems through dynamic heterogeneous redundancy characteristics of mimic systems, providing endogenous security mechanisms for information systems and offering a solution for enhancing security and reliability.

6 Conclusion

TSN technology has been applied in automotive networks, industrial control, professional A/V, and communication networks, achieving deterministic data stream transmission while leveraging traditional Ethernet' s high bandwidth and low cost. TSN enables time-sensitive data streams and best-effort data streams to transmit on the same medium, simplifying numerous devices and interfaces and gaining tremendous industry attention. However, TSN is a relatively new technology requiring continued improvement and innovation in many aspects, necessitating collaborative efforts between academia and industry. This paper introduced TSN' s development history, analyzed its key technologies in detail, elaborated on current research status, and provided future development prospects. It is hoped that this summary and organization can provide reference and assistance for researchers in related fields.

References

[1] Huang Tao, Wang Shuo, Huang Yudong, et al. Review of deterministic network research [J]. Journal on Communications, 2019, 40 (06): 160-176.

[2] Farkas J, Bello L, Gunther C. Time-sensitive networking standards [J]. IEEE Communications Standards Magazine, 2018, 2 (2): 20-21.

[3] Galloway B, Hancke G P. Introduction to industrial control networks [J]. IEEE Communications Surveys and Tutorials, 2013, 15 (2): 860-880.

[4] Shafi M, Molisch A F, Smith P J, et al. 5G: a tutorial overview of standards, trials, challenges, deployment, and practice [J]. IEEE Journal on Selected Areas in Communications, 2017, 35 (6): 1201-1221.

[5] Makowitz R, Temple C. FlexRay-a communication network for automotive control systems [C]// Proc of 2006 IEEE International Workshop on Factory Communication Systems. Piscataway, NJ: IEEE Press, 2006: 207-212.

[6] Committee A. ARINC specification 664P7: aircraft data network, part 7: Avionics Full-Duplex Switched Ethernet (AFDX) network Aeronautical Radio Inc, 2005.

[7] Feld J. PROFINET-Scalable factory communication for all applications [C]// Proc of 2004 IEEE International Workshop on Factory Communication Systems. Piscataway, NJ: IEEE Press, 2004: 33-38.

[8] Langlois K, Van Der Hoeven T, Cianca D R, et al. EtherCAT tutorial an introduction for real-time hardware communication on windows [J]. IEEE Robotics & Automation Magazine, 2018, 25 (1): 22-122.

[9] Cena G, Seno L, Valenzano A, et al. Performance analysis of Ethernet Powerlink networks for distributed control and automation systems [J]. Computer Standards & Interfaces, 2009, 31 (3): 566-572.

[10] Alvarez I, Proenza J, Barranco M, et al. Towards a time redundancy mechanism for critical frames in time-sensitive networking [C]// Proc of 2017 IEEE International Conference on Emerging Technologies and Factory Automation. Piscataway, NJ: IEEE Press, 2017: 1-4.

[11] Alvarez I, Barranco M, Proenza J, et al. Towards a fault-tolerant architecture based on time sensitive networking [C]// Proc of 2018 IEEE International Conference on Emerging Technologies and Factory Automation. Piscataway, NJ: IEEE Press, 2018: 1113-1116.

[12] Alvarez I, Almeida L, Proenza J, et al. A first qualitative comparison of the admission control in FTT-SE, HaRTES and AVB [C]// Proc of 2016 IEEE World Conference on Factory Communication Systems. Piscataway, NJ: IEEE Press, 2016: 1-4.

[13] IEEE 802.1Qcp-2018. Standard for Local and metropolitan area networks-Bridges and Bridged Networks-Amendment 30: YANG Data Model [S]. Piscataway, NJ: IEEE Press, 2018.

[14] IEEE 802.1Qav-2009. Standard for Local and metropolitan area networks-Virtual Bridged Local Area Networks Amendment 12: Forwarding and Queuing Enhancements for Time-Sensitive Streams [S]. Piscataway, NJ: IEEE Press, 2009.

[15] IEEE 802.1Qat-2010. Standard for Local and metropolitan area networks-Virtual Bridged Local Area Networks Amendment 14: Stream Reservation Protocol (SRP) [S]. Piscataway, NJ: IEEE Press, 2010.

[16] IEEE 802.1Qca-2015. Standard for Local and metropolitan area networks-Bridges and Bridged Networks-Amendment 24: Path Control and Reservation [S]. Piscataway, NJ: IEEE Press, 2016.

[17] IEEE 802.1Qbv-2015. Standard for Local and metropolitan area networks-Bridges and Bridged Networks-Amendment 25: Enhancements for Scheduled Traffic [S]. Piscataway, NJ: IEEE Press, 2016.

[18] IEEE 802.1Qbu-2016. Standard for Local and metropolitan area networks-Bridges and Bridged Networks-Amendment 26: Frame Preemption [S]. Piscataway, NJ: IEEE Press, 2016.

[19] IEEE 802.1Qch-2017. Standard for Local and metropolitan area networks-Bridges and Bridged Networks-Amendment 29: Cyclic Queuing and Forwarding [S]. Piscataway, NJ: IEEE Press, 2017.

[20] IEEE 8021Qci-2017. Standard for Local and metropolitan area networks-Bridges and Bridged Networks-Amendment 28: Per-Stream Filtering and Policing [S]. Piscataway, NJ: IEEE Press, 2017.

[21] IEEE 802.1Qcc-2018. Standard for Local and Metropolitan Area Networks-Bridges and Bridged Networks-Amendment 31: Stream Reservation Protocol (SRP) Enhancements and Performance Improvements [S]. Piscataway, NJ: IEEE Press, 2018.

[22] IEEE 802.1BA-2011. Standard for Local and metropolitan area networks-Audio Video Bridging (AVB) Systems [S]. Piscataway, NJ: IEEE Press, 2011.

[23] IEEE 802.1AS-2011. Standard for Local and Metropolitan Area Networks-Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks [S]. Piscataway, NJ: IEEE Press, 2011.

[24] IEEE 802.1CB-2017. Standard for Local and metropolitan area networks-Frame Replication and Elimination for Reliability [S]. Piscataway, NJ: IEEE Press, 2017.

[25] IEEE 802.1CM-2018. Standard for Local and metropolitan area networks-Time-Sensitive Networking for Fronthaul [S]. Piscataway, NJ: IEEE Press, 2018.

[26] IEEE 1588-2008 (Revision of IEEE Std 1588-2002). Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems [S]. Piscataway, NJ: IEEE Press, 2008.

[27] Zheng Hao. Design and implementation of low latency time-aware system based on IEEE 802.1AS [D]. Xi' an: Xidian University, 2018.

[28] Garner G, Ryu H. Synchronization of audio/video bridging networks using IEEE 802.1AS [J]. IEEE Communications Magazine, 2011, 49 (2): 140-147.

[29] IEEE 802.3br-2016. Standard for Ethernet Amendment 5: Specification and Management Parameters for Interspersing Express Traffic [S]. Piscataway, NJ: IEEE Press, 2016.

[30] Lo Bello L, Steiner W. A perspective on IEEE time-sensitive networking for industrial communication and automation systems [J]. Proceedings of the IEEE, 2019, 107 (6): 1094-1120.

[31] Lee J, Park S. Time-Sensitive Network (TSN) Experiment in Sensor-Based Integrated Environment for Autonomous Driving [J]. Sensors, 2019, 19 (5): 1111-1122.

[32] Li Yangang, Deng Wenping, Wang Hong, et al. Research and analysis on the differences between intra-domain routing protocols OSPF and IS-IS [J]. Computer Science, 2015, 42 (S1): 256-259.

[33] Craciunas S, Oliver S, Chmelik M, et al. Scheduling real-time communication in IEEE 802.1Qbv time sensitive networks [C]// Proc of 2016 International Conference on Real-Time Networks and Systems. New York, NY: ACM, 2016: 183-192.

[34] Oliver S, Craciunas S, Steiner W. IEEE 802.1Qbv gate control list synthesis using array theory encoding [C]// Proc of 2018 IEEE Real-Time and Embedded Technology and Applications Symposium. Piscataway, NJ: IEEE Press, 2018: 13-24.

[35] Ojewale M A, YOMSI P M. Routing heuristics for load-balanced transmission in TSN-based networks [J]. ACM Sigbed Review, 2020, 16 (4): 20-25.

[36] Pahlevan M, Obermaisser R. Genetic algorithm for scheduling time-triggered traffic in time-sensitive networks [C]// Proc of 2018 IEEE Inter-

national Conference on Emerging Technologies and Factory Automation. Piscataway, NJ: IEEE Press, 2018: 337-344.

[37] Falk J, Durr F, Rothermel K, et al. Exploring practical limitations of joint routing and scheduling for TSN with ILP [C]// Proc of 2018 IEEE International Conference on Embedded and Real-Time Computing Systems and Applications. Piscataway, NJ: IEEE Press, 2018: 136-146.

[38] Mahfouzi R, Aminifar A, Samii S, et al. Stability-aware integrated routing and scheduling for control applications in ethernet networks [C]// Proc of 2018 IEEE Design, Automation & Test in Europe Conference & Exhibition. Piscataway, NJ: IEEE Press, 2018: 682-687.

[39] Gavrilut V, Zhao L, Raagaard M, et al. AVB-aware routing and scheduling of time-triggered traffic for TSN [J]. IEEE Access, 2018, 6: 682-687.

[40] Zhao X, Pop P, Zheng Z, et al. Timing analysis of avb traffic in tsn networks using network calculus [C]// Proc of 2018 IEEE Real-Time and Embedded Technology and Applications Symposium. Piscataway, NJ: IEEE Press, 2018: 25-36.

[41] Zhang J, Chen L, Wang T, et al. Analysis of TSN for industrial automation based on network calculus [C]// Proc of 2019 IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). Piscataway, NJ: IEEE Press, 2019: 240-247.

[42] Benammar N, Bauer H, Ridouard F, et al. Timing analysis of AVB Ethernet network using the forward end-to-end delay analysis [C]// Proc of 2018 International Conference on Real-Time Networks and Systems. New York, NY: ACM, 2018: 223-233.

[43] Rivera-Verduzco H J, Cuijpers L, Cao J Y, et al. Work in progress: best-case response time analysis for Ethernet AVB [C]// Proc of 2017 IEEE Real-Time Systems Symposium. Piscataway, NJ: IEEE Press, 2017: 378-381.

[44] Maxim D, Song Y Q. Delay analysis of avb traffic in time-sensitive networks (TSN) [C]// Proc of 2017 International Conference on Real-Time Networks and Systems. New York, NY: ACM, 2017: 18-27.

[45] Zhang P, Liu Y, Shi J Q, et al. A feasibility analysis framework of time-sensitive networking using real-time calculus [J]. IEEE Access, 2019, 7: 682-687.

[46] Tan T Y, Park S K. Redundancy path implementation for schedule traffic [C]// Proc of 2018 IEEE International Workshop on Advanced Image Technology. Piscataway, NJ: IEEE Press, 2018: 1-3.

[47] Qian S F, Luo F, Xu J P, et al. An Analysis of Frame Replication and Elimination for Time-Sensitive Networking [C]// Proc of 2017 International Conference on Network, Communication and Computing. New York, NY: ACM, 2017: 166-170.

[48] Pahlevan M, Obermaisser R. Redundancy management for safety-critical applications with time sensitive networking [C]// Proc of 2018 IEEE International Telecommunication Networks and Applications Conference. Piscataway, NJ: IEEE Press, 2018: 1-7.

[49] Smirnov F, Reimann F, Teich J, et al. Automatic optimization of redundant message routings in automotive networks [C]// Proc of the 2018 International Workshop on Software and Compilers for Embedded Systems. New York, NY: ACM, 2018: 90-99.

[50] Gavrilut V, Zarrin B, Pop P, et al. Fault-tolerant topology and routing synthesis for IEEE time-sensitive networking [C]// Proc of 2017 International Conference on Real-Time Networks and Systems. New York, NY: ACM, 2017: 267-276.

[51] CPRI alliance V4.2. Common Public Radio Interface (CPRI) SPECIFICATION V4.2 [S]. 2010.

[52] Chitimalla D, Kondepu K, Valcarenghi L, et al. 5G fronthaul-latency and jitter studies of CPRI over Ethernet [J]. Journal of Optical Communications and Networking, 2017, 9 (2): 172-182.

[53] Hisano D, Nakayama Y, Kubo T, et al. Gate-shrunk time aware shaper: low-latency converged network for 5g fronthaul and m2m services [C]// Proc of IEEE Global Telecommunications Conference (GLOBECOM). Piscataway, NJ: IEEE Press, 2017: 1-6.

[54] Bjornstad S, Chen D, Veisllari R, et al. Handling delay in 5G Ethernet mobile fronthaul networks [C]// Proc of the 2018 IEEE European Conference on Networks and Communications. Piscataway, NJ: IEEE Press, 2018: 1-9.

[55] Nayak N G, Durr F, Rothermel K, et al. Time-sensitive software-defined network (TSSDN) for real-time applications [C]// Proc of 2016 International Conference on Real-Time Networks and Systems. New York, NY: ACM, 2016: 193-202.

[56] Hackel T, Meyer P, Korf F, et al. Software-defined networks supporting time-sensitive in-vehicular communication [C]// Proc of the 2019 IEEE vehicular technology conference (VTC Spring). Piscataway, NJ: IEEE Press, 2019: 1-5.

[57] Said S B H, Truong Q H, Boc M. SDN-based configuration solution for IEEE 802.1 time sensitive networking (TSN) [J]. SIGBED Rev, 2019, 16 (1): 27–32.

[58] Haur N K, Chin T S. Time-sensitive-aware scheduling traffic (tsa-st) algorithm in software-defined networking [C]// Proc of the 2019 International Conference on Internet and Distributed Computing Systems. Berlin, German: Springer, 2019: 248-259.

[59] Thiele D, Ernst R. Formal analysis based evaluation of software defined networking for time-sensitive ethernet [C]// Proc of the 2016 Design, Automation

& Test in Europe Conference & Exhibition. New York, NY: ACM, 2016: 31-36.

[60] Jia Z Y, Wang J L, Chen X, et al. An SDN-based measurement scheme to build delay database for time-sensitive network scheduling [J]. International Journal of Innovative Computing Information and Control, 2019, 15 (4): 1271-1286.

[61] Pruski A, Berger M. Design considerations for high-performance Time sensitive networking switches [C]// Proc of the 2019 10th International Conference on Networks of the Future (NoF). Piscataway, NJ: IEEE Press, 2019: 114-117.

[62] Jin X, Xia C, Guan N, et al. Real-time scheduling of massive data in time sensitive networks with a limited number of schedule entries [J]. IEEE Access, 2020, 8: 6751-6767.

[63] Ziebinski A, Cupek R, Grzechca D, et al. Review of advanced driver assistance systems (ADAS) [C]// Proc of the AIP Conference Proceedings, New York: AIP, 2017: 1-4.

[64] Ti Fang. The research of stream reservation protocol for in-vehicular time-sensitive network [D]. Xi' an: Xidian University, 2018.

[65] Liao Y, Deschamps F, Loures E, et al. Past, present and future of Industry 4.0-a systematic literature review and research agenda proposal [J]. International journal of production research, 2017, 55 (12): 3609-3629.

[66] Lyu Ping, Liu Qinrang, Wu Jiangxing, et al. Next-generation software-defined architecture [J]. SCIENTIA SINICA Informationis, 2018, 48 (03): 315-28.

[67] Wu Jiangxing. Introduction to cyberspace mimic defense [M]. Beijing: Science Press, 2016.

Note: Figure translations are in progress. See original paper for figures.