A Survey of Real-Time Ethernet Modeling and Design Methodologies: From AVB to TSN

LIBING DENG and GUOQI XIE, Hunan University
HONG LIU, Shanghai Trusted Industrial Control Platform Co., Ltd.
YUNBO HAN, Tencent CSIG
RENFA LI, Hunan University
KEQIN LI, State University of New York

With the development of real-time critical systems, the ever-increasing communication data traffic puts forward high-bandwidth and low-delay requirements for communication networks. Therefore, various real-time Ethernet protocols have been proposed, but these protocols are not compatible with each other. The IEEE 802.1 Working Group developed standardized protocols named Audio Video Bridging (AVB) in 2005, and renamed it Time-Sensitive Networking (TSN) later. TSN not only adds new features but also retains the original functions of AVB. Proposing real-time Ethernet modeling and design methodologies is the key to meeting high-bandwidth and low-delay communication requirements. This article surveys the modeling from AVB to TSN, mainly including: (1) AVB and TSN modeling; (2) end-to-end delay modeling; (3) real-time scheduling modeling; (4) reliability modeling; and (5) security modeling. Based on these models, this article surveys the recent advances in real-time Ethernet design methodologies from AVB to TSN: (1) end-to-end delay analysis from AVB to TSN; (2) real-time scheduling from AVB to TSN; (3) reliability-aware design for TSN; and (4) security-aware design for TSN. Among the above four points, the last two points are only for TSN, because AVB lacks reliability and security mechanisms. This article further takes the automotive use case as an example to discuss the application of TSN in automobiles. Finally, this article discusses the future trends of TSN. By surveying the recent advances and future trends, we hope to provide references for researchers interested in real-time Ethernet modeling and design methodologies for AVB and TSN.

CCS Concepts: • **Networks → Cyber-physical networks**; **Network performance modeling;**

Additional Key Words and Phrases: Audio Video Bridging (AVB), real-time Ethernet, Time-Sensitive Networking (TSN)

This work was supported in part by the National Natural Science Foundation of China under Grants No. 61932010, No. 61972139, and No. 62133014, the Opening Project of Shanghai Trusted Industrial Control Platform, and the Fundamental Research Funds for the Central Universities, Hunan University, China.

Authors' addresses: L. Deng and R. Li, Key Laboratory for Embedded and Network Computing of Hunan Province, College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China; emails: {libingdeng, lirenfa}@hnu.edu.cn; G. Xie (corresponding author), Key Laboratory for Embedded and Network Computing of Hunan Province, College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, Center for Convergence of Automobile and Cyberspace, Research Institute of Hunan University in Chongqing, Chongqing 401120, China; email: xgqman@hnu.edu.cn; H. Liu, Shanghai Trusted Industrial Control Platform Co. Ltd., Shanghai 200062, China; email: hliu@sei.ecnu.edu.cn; Y. Han, Tencent CSIG, Shenzhen 518057, China; email: yunbohan@tencent.com; K. Li, State University of New York, New Paltz 12561; email: lik@newpaltz.edu.

0360-0300/2022/01-ART31 $15.00
https://doi.org/10.1145/3487330

ACM Reference format:
Libing Deng, Guoqi Xie, Hong Liu, Yunbo Han, Renfa Li, and Keqin Li. 2022. A Survey of Real-Time Ethernet Modeling and Design Methodologies: From AVB to TSN. *ACM Comput. Surv.* 55, 2, Article 31 (January 2022), 36 pages.
https://doi.org/10.1145/3487330

1 INTRODUCTION

1.1 Background

With the development of real-time critical systems such as automobiles, aerospace, industrial automation, and industrial control, the ever-increasing communication data traffic puts forward high-bandwidth and low-delay requirements for communication networks. However, traditional industrial field bus cannot meet the above communication requirements of real-time critical systems. For example, intelligent and autonomous driving systems, such as **Advanced Driver Assistant System** (**ADAS**), **Adaptive Cruise Control** (**ACC**), and **Adaptive Front-lighting System** (**AFS**), are increasingly developed and applied in automobiles [19]. As a result, the data traffic and the number of **Electronic Control Units** (**ECUs**) continue to increase, thereby putting forward high-bandwidth and hard real-time requirements for automotive networks [30]. However, traditional in-vehicle buses (e.g., CAN, FlexRay, MOST, and so on.) cannot meet the above communication requirements. Therefore, various companies have proposed different real-time Ethernet protocols, such as EtherCAT [74], ProfiNet [109], and POWERLINK [27]. These real-time Ethernet protocols meet the real-time and determinacy communication requirements in individual fields. In the age of Industry 4.0, the application software based on **Information Technology** (**IT**) needs to interact with the on-site **Operational Technology** (**OT**) information. However, the above real-time Ethernet protocols cannot solve the problem of information integration between IT and OT. The main reasons include: (1) it is difficult to standardize individual real-time Ethernet protocols due to the competition among different manufacturers; (2) these real-time Ethernet protocols are incompatible with each other, making it impossible to achieve interoperability among devices in different real-time networks; and (3) partial real-time Ethernet products need to use dedicated ASIC chips, making high-development cost and maintenance cost.

In 2005, the IEEE 802.1 Working Group[57] built the unified data link layer protocol named **Audio Video Bridging** (**AVB**). AVB ensures the real-time transmission of audio and video streams, which makes it suitable for automobile, avionics industry, and other industrial fields with high-bandwidth and low-delay requirements. In addition to the real-time requirement, reliability should also be considered in real-time critical systems. System failures may result in high economic losses and even endanger human life. However, AVB lacks fault-tolerant mechanisms to enhance its reliability [62]. At the same time, in the face of malicious cyber attacks, AVB lacks security protection mechanism for real-time critical systems [114]. Therefore, the IEEE 802.1 Working Group expanded AVB and renamed it as **Time-Sensitive Networking** (**TSN**) [59] in 2012. TSN has the following new features compared with AVB: (1) TSN further enhances the time synchronization, improves the streams reservation ability, and supports the scheduling of time-sensitive data streams. In other words, TSN further emphasizes the hard real-time requirement of time-sensitive data; (2) TSN adds redundancy functions that include the frame replication and elimination, the path control, and reservation ability, thereby providing high-reliability guarantee mechanisms; and (3) TSN adds stream filtering and policing function, thereby providing the security guarantee strategy for data transmission.

TSN can be operated isomorphically through standardization in different fields, thereby improving the connectivity and versatility of different devices in real-time critical systems. TSN further guarantees communication requirements of hard real-time, reliability, and security, making it suitable for real-time critical systems while meeting the integration of IT and OT.

1.2 Motivation

AVB and TSN can meet the communication requirements of hard real-time, reliability, and security for real-time critical systems, where AVB only guarantees the real-time performance (although AVB is expanded and renamed to TSN, AVB has not been replaced and is still used in real-time critical systems such as automobiles and aviation systems). Therefore, research on AVB and TSN has important practical significance. Ref. [71] used simulation to evaluate the performance of AVB protocols. Ref. [28] introduced the principles of AVB protocols, such as time synchronization, bandwidth allocation; it further discussed the application of AVB in live and introduced the progress of the next generation AVB. Refs. [12, 114] considered the application prospects of AVB and TSN in the automotive and avionics industry, respectively. Ref. [76] presented related concepts and protocols of TSN and discussed the application of TSN in the automotive industry. Ref. [20] focused on data scheduling mechanisms and methods for TSN. Ref. [104] summarized TSN related protocols and technologies based on the application of TSN in automotive systems.

Since AVB and TSN protocols are theoretical, it is necessary to modify and design the protocol according to actual requirements in application scenarios to improve system performance. Therefore, the modeling and design methodologies are crucial. However, the above reviews from AVB to TSN mainly focus on protocols, related mechanisms, and applications, but do not consider AVB and TSN modeling and design methodologies. In order to apply AVB and TSN protocols to different fields, and to meet communication requirements (i.e., hard real-time, reliability, and security), it is essential to propose effective modeling and design methodologies for AVB and TSN, through which AVB and TSN can meet the above communication requirements in concrete fields of industries. However, modeling and design methodologies about real-time, reliability, and security from AVB to TSN still face the following challenges.

(1) The end-to-end delay of communication messages is the basis for ensuring the safe operation of the system. With the rapid development of real-time critical systems (e.g., automobile, avionics, and industrial automation), the physical environment of systems becomes increasingly complex and the type and number of data streams in the system network are gradually increasing. These factors increase the difficulty of calculating the end-to-end delay of communication messages in AVB network or TSN. However, the analysis methods of end-to-end delay include model-checking method [84] and simulation [24]; the model-checking method needs to cover all scenarios, and the simulation requires a lot of time and effort during model building and execution. The above methods are inefficient for middle-scale or large-scale networks. Mathematical analysis method can cover all possible scenarios without spending a lot of time through mathematical modeling, but it exists a certain pessimism. At present, the mathematical analysis method that calculates the end-to-end delay for AVB and TSN messages has made great progress (see Section 3 for details).

(2) In communication network of real-time critical systems, efficient scheduling algorithms can transmit data in time. Considering that different types of messages (e.g., critical traffic and non-critical traffic) are transmitted in AVB network and TSN, we should use different scheduling algorithms to avoid data collision, thereby guaranteeing the **Quality of Service** (**QoS**) of various types of messages (especially for hard real-time messages). The scheduling algorithms can realize the deterministic transmission of real-time messages, and guarantee the reliable and safe operation of the system. Real-time scheduling algorithms from AVB to TSN communications have made great breakthroughs (see Section 4 for details).

(3) Real-time critical systems require not only the real-time performance of the network, but also the reliability of the network. Redundant mechanisms can allow the network to

operate normally when a network failure occurs, thereby improving system reliability. However, AVB does not provide redundant communication capacity, such that it is difficult to guarantee the reliability of AVB network systems [62]. The emergence of TSN solves this problem, many researches have been carried out on the reliability-aware design for TSN (see Section 5 for details).

(4) The integration of IT and OT makes frequent interaction between the system and external environment, which increases the risk of the system being attacked by external environment and may endanger personal and property safety. Therefore, it is necessary to ensure the security of the system. However, AVB lacks the security mechanism, such that it is not suitable for real-time critical systems. TSN considers security from the aspects of data streams filtering and management, and some new progress has been made (see Section 6 for details).

1.3 Contributions and Outline

In view of the above challenges, this article surveys related models, recent advances, and future trends from AVB to TSN. The contributions of this article are as follows.

(1) Modeling from AVB to TSN: This article surveys AVB and TSN models, mainly including: (1) AVB and TSN network system modeling; (2) end-to-end delay modeling; (3) real-time scheduling modeling; (4) reliability modeling; and (5) security modeling. Based on these models, different methods are designed for applying AVB and TSN to the concrete fields of industries.

(2) End-to-end delay analysis: For real-time critical applications, the real-time requirement can be verified by the **worst-case end-to-end delay** (**WCD**). Mathematical analysis methods (e.g., formal analysis and network calculus) solve the problem of WCD calculation of AVB and TSN messages under different interference factors. In this article, the end-to-end delay analysis methods from AVB to TSN are provided to realize real-time communication.

(3) Real-time scheduling: Real-time scheduling applies different scheduling algorithms for different types of traffic in communication networks. At the same time, scheduling algorithms further control the forwarding and transmission of data traffic by combining routing optimization, thereby achieving deterministic transmission of data streams in the network. This article integrates the real-time scheduling algorithms from AVB to TSN to provide the real-time guarantee and reliable operation of the system.

(4) Reliability-aware design: Different redundancy methods for different failure types are designed to ensure the reliability of the network. The design methods surveyed in this article provide a reference solution for network designers to design reliable networks. As AVB lacks the redundancy mechanism, this article provides recent reliability-aware design methods (using fault tolerance) for TSN failures.

(5) Security-aware design: The security-aware design methods (e.g., network data streams management and information encryption) ensure cyber security. As AVB does not provide security guarantees, this article surveys the security-aware design methods for ensuring the security communication in TSN.

(6) TSN in automotive domain: This article takes the automotive use case as an example to discuss the application of TSN in automobiles, including: (1) TSN-based automotive E/E architecture; (2) end-to-end delay analysis for TSN-based automotive E/E architecture; (3) real-time scheduling for TSN in automobiles; and (4) P802.1DG—TSN profile for automobiles. The above contents provide the current status, challenges, and trends of TSN in automotive domain.

(7) Future trends: This article further discusses the future trends of TSN, mainly including: (1) automated configuration of TSN; (2) security enhancement of TSN; and (3) congestion control of TSN. By introducing the future trends, we provide the development prospects and the directions that require in-depth research of TSN.

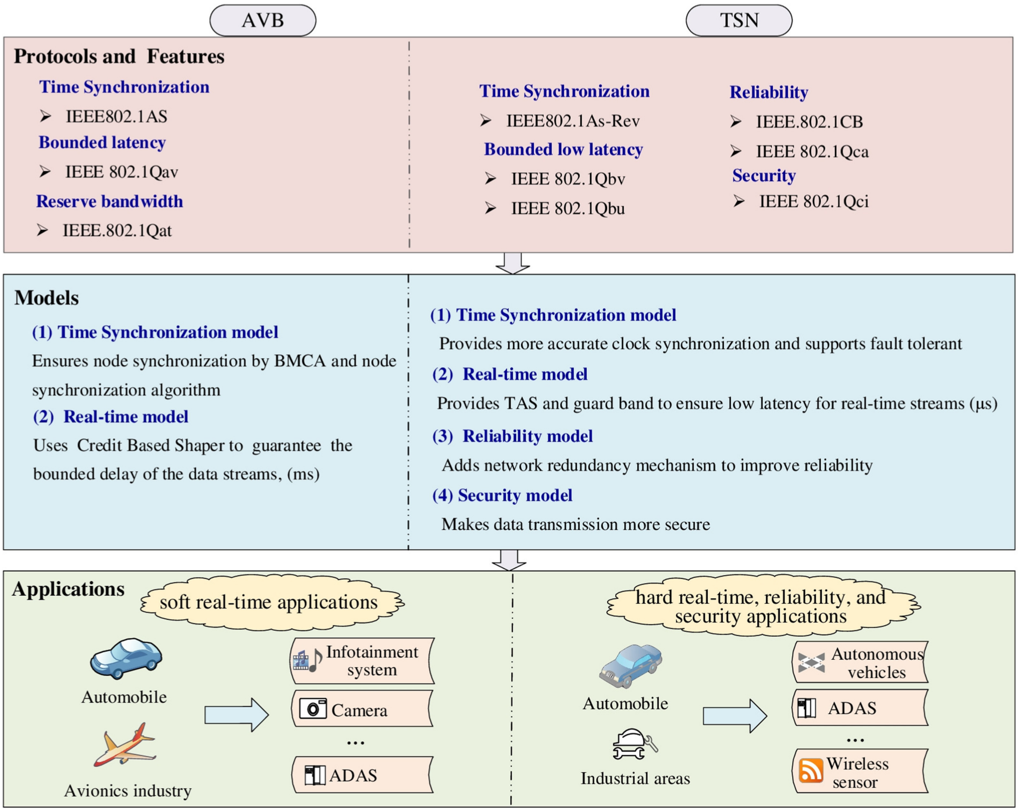

Fig. 1. Comparison between AVB and TSN.

This article surveys the real-time analysis and design from AVB to TSN through two aspects: (1) end-to-end delay analysis; (2) real-time scheduling. The reliability-aware design methods and security-aware design methods provide safety enhancement and security enhancement measures, respectively.

The outline of this article is below. Section 2 surveys and compares AVB and TSN in terms of protocols, models, and applications. Sections 3–6 survey the recent advances of end-to-end delay analysis, real-time scheduling, reliability-aware design, and security-aware design. Section 7 discusses the application of TSN in automotive domain. Section 8 discusses the future trends of TSN, and Section 9 concludes this article.

2 MODELING FROM AVB TO TSN

The related models are established according to the protocols and features of AVB and TSN, so that AVB and TSN can be applied in different fields. Therefore, this section mainly surveys and compares AVB and TSN in terms of: (1) network protocols and their features; (2) models; and (3) applications, as shown in Figure 1.

2.1 Network Protocols and Features from AVB to TSN

AVB meets the requirements of real-time messages such as audio and video streams. In AVB, real-time messages transmitted in time should meet: (1) devices connected in the network need to have

Table 1. The Main Protocols of AVB and TSN

AVB/TSN	Name	Function	Feature
AVB	802.1AS [48]	generalized Precision Time Protocol (gPTP)	real-time
AVB	802.1Qav [52]	Forwarding and Queueing of Time-Sensitive Streams (FQTSS)	real-time
AVB	802.1Qat [51]	Stream Reservation Protocol (SRP)	real-time
TSN	802.1AS-REV [49]	Time synchronization (enhanced 802.1AS)	real-time
TSN	802.1Qbv [54]	Enhanced traffic scheduling based on 802.1Qav	real-time
TSN	802.1Qbu [53]	Frame preemption	real-time
TSN	802.1CB [50]	Frame replication and elimination	reliability
TSN	802.1Qca [55]	Path control and reservation	reliability
TSN	802.1Qci [56]	Per-stream filtering and policing	security

synchronized wall clock time; (2) data streams must be transmitted before the maximum time delay. Therefore, the IEEE 802.1 Working Group formulated AVB protocols based on the traditional IEEE 802.1Q, mainly including the protocols listed in Table 1. AVB mainly provides the following three capabilities to ensure the transmission of real-time messages.

(1) Time synchronization: 802.1AS **generalized Precision Time Protocol** (**gPTP**) ensures strict time synchronization among network nodes, which is the basis for timely response between nodes.

(2) Bounded latency: 802.1Qav **Forwarding and Queueing of Time-Sensitive Streams** (**FQTSS**) achieves the goal of low delay by providing data streams processing and forwarding services, such as the **Credit-based Shaper** (**CBS**), prioritization, and queue management.

(3) Bandwidth reservation: 802.1Qat **Stream Reservation Protocol** (**SRP**) guarantees the end-to-end delay of AVB traffic by reserving bandwidth for AVB traffic (i.e., SR_A and SR_B).

Although AVB has many advantages, it still cannot meet the application requirements in real-time critical systems, mainly including the following three aspects.

(1) Hard real-time: For example, applications of ADAS (e.g., sensors and radars) in automobiles require more timely response time (microsecond) to ensure the real-time performance. AVB realizes the end-to-end delay (2 ms) over 7 hops, so it is not suitable for these applications.

(2) Reliability: For applications with extremely reliability and security requirements, AVB lacks the good fault recovery mechanism and cannot guarantee system reliability.

(3) Security: AVB does not provide network security mechanisms to ensure the confidentiality and integrity of data transmission in the network.

In order to meet the above requirements, TSN improves to AVB and proposes some new standards, as shown in Table 1. Compared with AVB, TSN protocols have the following features.

(1) More accurate time synchronization: TSN provides a simpler and faster time synchronization mechanism (802.1AS-REV), further improves the synchronization accuracy, and supports redundancy mechanism.

(2) Hard real-time: TSN uses stronger traffic scheduling function (802.1Qbv) and frame preemption mechanism (802.1Qbu) to provide low delay and jitter guarantee for time-sensitive data; in other words, TSN further emphasizes on the hard real-time requirement of time-sensitive data.

(3) Reliability: TSN adds redundant strategies (802.1CB and 802.1Qca) to realize the fast recovery of network failures.

(4) Security: TSN adds stream filtering and policing function (802.1Qci) to make data transmission more secure.

2.2 AVB and TSN Modeling

In order to understand the architecture and application of AVB and TSN, we summarize AVB and TSN models, including network architecture and application models.

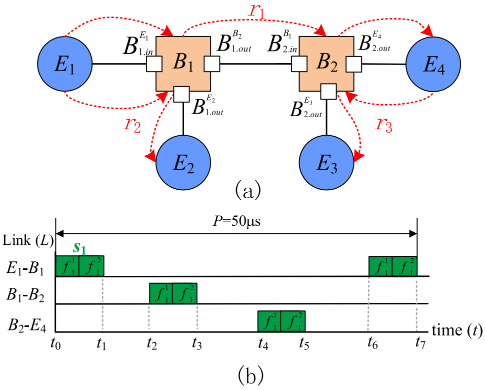

Fig. 2. AVB and TSN network architecture.

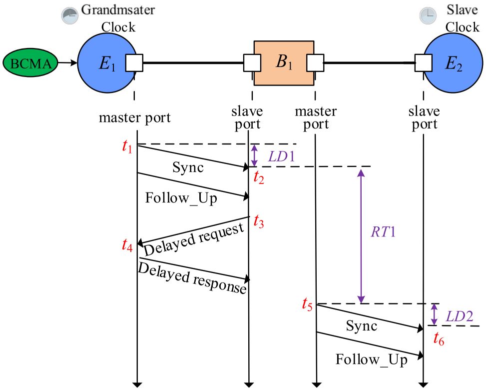

Fig. 3. Time synchronization modeling.

Table 2. The Symbols in AVB and TSN Architecture Model

Symbol	**Meaning**	**Examples**
E	end device	E_1, E_2, E_3, E_4
B	bridge	B_1, B_2
$B.^{decice}_{in}$	in-port of bridge connected to the device	$B^{E_1}_{1.in}$, $B^{B_1}_{2.in}$
$B.^{decice}_{out}$	out-port of bridge connected to the device	$B^{E_2}_{1.out}$, $B^{E_4}_{2.out}$
r	route	r_1, r_2, r_3
L	data transmission link	E_1-B_1, B_1-B_2

(1) AVB and TSN network architecture. AVB and TSN network architecture are composed of bridges (also called switches) and end devices that support AVB and TSN protocols, and the bridges and end devices are connected by physical links. A network architecture model with four end devices (E_1, E_2, E_3, E_4) and two bridges (B_1, B_2) is shown in Figure 2(a); where the blue circle represents the end device; the orange square represents the bridge, each bridge contains in-port (also called ingress port, port in, incoming port, input port, internal port) and out-port (also called egress port, port out, outgoing port, output port, external port), such as $B^{E_1}_{1.in}$ and $B^{E_2}_{1.out}$. $B^{E_1}_{1.in}$ represents that E_1 is connected to B_1 through the in-port $B_{1.in}$ and $B^{E_2}_{1.out}$ represents that E_2 is connected to B_1 through the out-port $B_{1.out}$; the black line represents the physical link (full-duplex) between the end device and the bridge; the red dotted line with the arrow represents the route (e.g., r_1, r_2, and r_3) passed during data steams transmission; the route is the directional path through which data is transmitted from the sending end device to the destination end device, such as r_1=(E_1-B_1, B_1-B_2, B_2-E_4).

In order to better understand AVB and TSN network architecture, Table 2 explains the symbols that appear in AVB and TSN architecture model in Figure 2.

(2) AVB and TSN application modeling. There are different types of data traffic in real-time network, mainly including **Control Traffic** (**CT**), AVB traffic (i.e., SR_A and SR_B: the priority of SR_A is higher than that of SR_B), and **Best-Effort traffic** (**BE**). All data streams (i.e., flows) in the network are represented as $S = \{S_{\text{CT}} \cup S_{\text{AVB\_A}} \cup S_{\text{AVB\_B}} \cup S_{\text{BE}}\}$. Table 3 lists the relevant attributes of the data stream s. For each stream $s_i \in S$, it consists of own data attributes and transmission attributes, expressed as $s_i = \{P_{\text{d}} \cup P_{\text{t}}\}$. P_{d} and P_{t} are explained as follows.

Table 3. The Attributes of Data Stream in Network

Symbol	**Meaning**	**Examples**
s	data stream	s_1
$size$	data size	1600 B
P	the period of data transmission	50 µs
D	the deadline of data transmission	50 µs
f_i^j	the jth frame belonging to the stream s_i	f_1^1, f_1^2
$fSet(s_i)$	the frames set of stream s_i	$fSet(s_1) = \{f_1^1, f_1^2\}$
$fNum$	the number of frames in a stream	the number of frames in the stream s_1 is 2
E_S	sending end of data stream	E_1
E_R	receiving end of data stream	E_4
r	route	$r_1 = (E_1\text{-}B_1, B_1\text{-}B_2, B_2\text{-}E_4)$

(1) The attribute P_{d} is related to the attribute of the data itself, mainly including the data size, transmission period, deadline, the number of frames, and the frame set of stream. These attributes are expressed as *size*, *T*, *D*, *fNum*, *fSet(s)*, respectively. Since the size of the message exceeds the **Maximum Transmission Unit** (**MTU**) of the frame, the message is divided into multiple frames, represented by

$$fSet(s_i) = \{f_i^1, f_i^2, \ldots, f_i^j, \ldots, f_i^n\},$$

where f_i^j in $fSet(s_i)$ represents the jth frame belonging to the stream s_i ($0 < j < n + 1$, j is an integer); then $fNum(s_i)$ is obtained by

$$fNum(s_i) = \lceil \frac{size}{\mathrm{MTU}} \rceil.$$

(2) The attribute P_{t} is related to the attribute of the data transmission process, mainly including the sending end, the receiving end, and the route. These attributes are denoted as E_S, E_R, and r, respectively. Figure 2(b) briefly shows the transmission process of network message, that is, the data stream s_1 containing frames f_1^1 and f_1^2 with 50µs period is sent from E_1 at t_0 along a transmission path r_1=(E_1-B_1, B_1-B_2, B_2-E_4) until E_4 is completely received at t_5. The data stream s_1 completes transmission before the deadline (50µs), so the stream s_1 is schedulable and meets real-time requirement.

2.3 End-to-End Delay Modeling from AVB to TSN

The basis of real-time analysis from AVB to TSN is to ensure the clock synchronization of each network node. Therefore, this article surveys the time synchronization model, through which the end-to-end analysis delay model is established.

(1) Time synchronization modeling from AVB to TSN. AVB proposes 802.1AS (gPTP), which improves **Precision Time Protocol** (**PTP**) specified in IEEE 1588 [66]. The protocol mainly realizes the time synchronization of network nodes from two aspects: (1) master clock selection; (2) node synchronization. Figure 3 shows the time synchronization model and its synchronization process.

(1) Master clock selection. 802.1AS selects the end device node in the network as the master clock (standard clock) by defining the **Best Master Clock Algorithm** (**BMCA**) [71], and other end device nodes as slave clocks (clocks that need to be proofread) to achieve network synchronization based on the Master clock. Figure 3 shows that the BMCA algorithm selects E_1 as the master clock and E_2 as the slave clock.

(2) Node synchronization. The clock synchronization accuracy of network nodes is mainly affected by the following factors: (1) link delay caused by data transmission on the link between two nodes; (2) the time consumed by the bridge in the process of forwarding data (i.e., residence time). We introduce the time synchronization process of network nodes in Figure 3.

The Master clock E_1 sends a Sync message at t_1 and B_1 receives the message at t_2. Subsequently, the Master clock E_1 sends a Follow_Up message with the *precisionOriginTimestamp* and *correctionField,* where *precisionOriginTimestamp* represents the time when the Master clock sends the synchronization message (i.e., t_1) and *correctionField* represents the offset between the local time and the Master clock sending the synchronization message. After receiving the Follow_Up message, B_1 forwards the synchronization message Sync at t_5. The offset between t_5 and t_1, the *correctionField*, includes the link delay (i.e., $LD1$ and $LD2$) between E_1 and E_2 and residence time ($RT1$) in B_1.

For link delay, gPTP proposes the peer-to-peer delay method to measure the link delay between two nodes and the link delay is symmetric. B_1 sends a Delay_Req message at t_3 after obtaining the t_1 value through the Follow_Up message. The Master clock E_1 sends the Delay_Resp with the t_4 value after receiving the message at t_4. The link delay between node E_1 and B_1 is obtained by using t_1, t_2, t_3, and t_4 values and is expressed by

$$LD1 = \frac{(t_4 - t_3 + t_2 - t_1)}{2}. \tag{1}$$

For residence time in the bridge, it is measured by the bridge. These two values (i.e., link delay and residence time) are accumulated in the *correctionField* for E_2 to synchronize with the Master clock E_1. The *correctionField* is expressed as

$$correctionField = RT1 + LD1, \tag{2}$$

where $RT1$ is residence time in the bridge. Therefore, the Master clock time of E_1 at t_6 is obtained by

$$MT(t_6) = t_1 + correctionField + LD2, \tag{3}$$

where MT is the Master clock time, and $LD2$ is calculated as $LD1$. The clock offset of E_2 is obtained by

$$offset = MT(t_6) - t_6. \tag{4}$$

E_2 adjusts and synchronizes its time to E_1 according to the *offset*. In order to meet the reliability requirement of network, 802.1AS-Rev adds the clock redundancy mechanism based on the principle of 802.1AS synchronization to ensure time synchronization performance when the Master clock fails. TSN also provides more accurate clock synchronization mechanism by supporting Sync message containing the time information t_1 without sending Follow_Up message.

Based on time synchronization of nodes in real-time network, the end-to-end delay analysis model is established.

(2) End-to-end delay analysis modeling. The end-to-end delay of data messages in AVB and TSN is the time interval that the message is sent from the sending end along the route until the receiving end is completely received, and is expressed by

$$ETE(m) = t.r - t.s, \tag{5}$$

where $ETE(m)$ represents the end-to-end delay of message m; $t.r$ represents the moment when the receiving end completely receives the message m, and $t.s$ represents the moment when the sending end starts to send the message m.

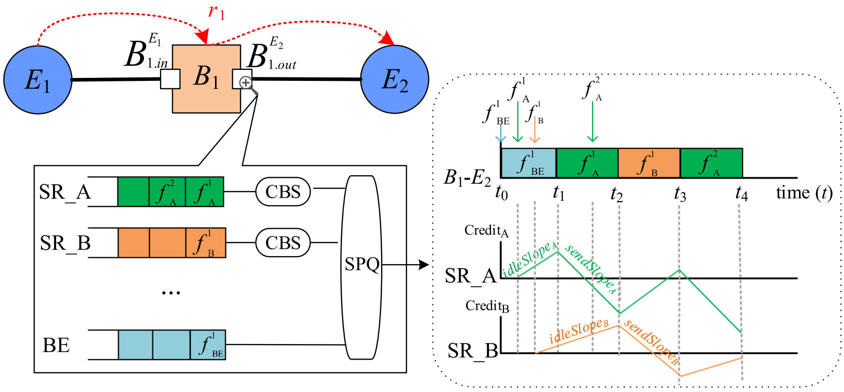

Fig. 4. Real-time scheduling modeling for AVB.

Figure 2(b) shows the transmission process of the message s_1; the message s_1 is sent from the sending end E_1 at t_0 until the receiving end E_4 is completely received at t_5. The message s_1 is transmitted along the route r_1=(E_1-B_1, B_1-B_2, B_2-E_4). The end-to-end delay of s_1 is obtained by

$$ETE(s_1) = t_5 - t_0.$$

Another important concept related to end-to-end delay is WCD, which is the maximum delay time for a message to be transmitted from the sending end to the receiving end along the transmission path (route), that is,

$$WCD(m) = max\{ETE(m)^1, ETE(m)^2, \ldots, ETE(m)^n\}, \tag{6}$$

where $WCD(m)$ represents the WCD of message m, $ETE(m)^n$ represents nth possible case of end-to-end delay for message m.

2.4 Real-Time Scheduling Modeling from AVB to TSN

(1) Real-time scheduling modeling for AVB. In terms of AVB message scheduling, AVB proposes CBS to ensure the real-time transmission of audio and video data streams. AVB traffic (SR_A and SR_B) is shaped through CBS to ensure transmission delay in AVB network; in addition, there are BE messages in AVB network. The forwarding process of data stream in AVB network is shown in Figure 4. For AVB traffic (the green part indicates the SR_A message and the orange part indicates the SR_B message) and the BE traffic (the light blue part indicates the BE message) in different queues, **Strict Priority Queuing** (**SPQ**) scheduling algorithm is adopted to transmit the message; for AVB traffic, the transmission delay is guaranteed by CBS to provide real-time data transmission services; for the messages in each queue, the **First Input First Output** (**FIFO**) method is used for scheduling.

The transmission of SR_A and SR_B messages must meet the conditions: (1) there is no higher message transmission than SR type traffic in AVB network; (2) the CBS stipulates that SR_A and SR_B messages will only be transmitted when the credit value is non-negative, and the credit value of SR_A and SR_B depends on bandwidth guarantee parameters: $idleSlope_{A/B}$ and $sendSlope_{A/B}$.

The relationship between $idleSlope_{A/B}$ and $sendSlope_{A/B}$ [1] is

$$sendSlope_{A/B} = idleSlope_{A/B} - portTransmitRate,$$

where the value of $idleSlope_{A/B}$ is the assigned bandwidth currently used by the queues related to traffic (SR_A or SR_B); $portTransmitRate$ is the maximum transmission data rate supported by the port, which is provided by the operation of the MAC Service.

The rules of credit value change are as follows. (1) The credit value is initially 0, and the data transmission is started when the credit is not negative; (2) The credit value will decrease with the *sendSlope* when the message is transmitting; (3) The credit value will increase with the *idleSlope* when the messages in the queue are waiting to transmit; and (4) When the credit value is positive and there is no message in the queue waiting to transmit, the credit value is reset to 0.

The detailed transmission process of messages frames (f_{BE}^1, f_{A}^1, f_{A}^2, and f_{B}^1) on link E_1-B_1 in AVB network is shown in Figure 4; where the downward arrow represents the arrival time of the message in the queue at the out-port $B_{1.out}^{E_2}$; the green and orange lines represent the credit value of SR_A and SR_B (i.e., $\mathrm{credit}_{\mathrm{A}}$ and $\mathrm{credit}_{\mathrm{B}}$), respectively.

(1) At time t_0, f_{BE}^1 begins to transmit; f_{A}^1 and f_{B}^1 arrive at the queue in $B_{1.out}^{E_2}$ during f_{B}^1 transmission, and the $\mathrm{credit}_{\mathrm{A}}$ and $\mathrm{credit}_{\mathrm{B}}$ begin to increase with $idleSlope_{\mathrm{A}}$ and $idleSlope_{\mathrm{B}}$, respectively.

(2) At time t_1, f_{BE}^1 transmission is completed, f_{A}^1 and f_{B}^1 are waiting to be transmitted in their respective queue, and the $\mathrm{credit}_{\mathrm{A}}$ and $\mathrm{credit}_{\mathrm{B}}$ are greater than 0; f_{A}^1 is transmitted since f_{A}^1 has a higher priority than f_{B}^1; during f_{A}^1 transmission, the $\mathrm{credit}_{\mathrm{A}}$ decreases with the $sendSlope_{\mathrm{A}}$, and the $\mathrm{credit}_{\mathrm{B}}$ continues to increase with the $idleSlope_{\mathrm{B}}$; meanwhile, f_{A}^2 arrives at the queue in $\mathrm{B}_{1.out}^{E_2}$ and waits for transmission.

(3) At time t_2, f_{A}^1 transmission is completed, and f_{A}^2 and f_{B}^1 are waiting to be transmitted in the queue; although f_{A}^2 has a higher priority than f_{B}^1, credit_A is negative, so that f_{B}^1 starts transmission; during f_{B}^1 transmission, the $\mathrm{credit}_{\mathrm{A}}$ increases with the $idleSlope_{\mathrm{A}}$ and the $\mathrm{credit}_{\mathrm{B}}$ decreases with the $sendSlope_{\mathrm{B}}$.

(4) At time t_3, f_{B}^1 transmission is completed, f_{A}^2 is waiting for transmission in the queue and the $\mathrm{credit}_{\mathrm{B}}$ is greater than 0. At this moment, f_{A}^2 transmission begins; during f_{A}^2 transmission, the credit_A decreases with the $sendSlope_{\mathrm{A}}$ and the $\mathrm{credit}_{\mathrm{B}}$ continues to increase with the $idleSlope_{\mathrm{B}}$.

AVB guarantees the end-to-end delay (2 ms) over 7 hops, but it is not suitable for more strict time-limited applications such as automotive control systems.

(2) Real-time scheduling modeling for TSN. TSN meets the hard real-time transmission requirement for CT message by defining **Time-Aware Shaper** (**TAS**), which assigns the **Gate Control List** (**GCL**) for each queue of each out-port to achieve data scheduling. The data starts to be transmitted only when the queue door is opened. There are eight queues (priority 0–7) at out-port, where priority 7 represents the highest priority queue and priority 0 represents the lowest priority queue. The message frame is selected to enter the corresponding queue for forwarding according to the **Priority Code Point** (**PCP**) of frame; PCP is included in the IEEE 802.1Q header of frame and is a three-bit field that supports eight different priority traffics.

For message frames in different queues at out-port, different scheduling algorithms are used for transmission and forwarding. For CT, TAS implements scheduling by specifying the GCL for each queue of each out-port, in which data can be transmitted when the door is opened (denoted by *o*); the data will wait to be transmitted in their respective queue when the door is closed (denoted by *c*). TAS introduces a guard band mechanism before CT transmission; the mechanism ensures that CT is not interfered by other traffic during transmission by closing other queue gates. For AVB traffic, CBS is used to ensure its low delay. For the messages in each queue, FIFO method is used for scheduling. For transmittable frames in different queues, SPQ scheduling algorithm is adopted to transmit high-priority traffic with priority.

The value of $idleSlope_{\mathrm{A/B}}$ will change due to the introduction of TAS. When gate is open, there is

$$idleSlope_{\mathrm{A/B}} = ActualBandwidth_{\mathrm{A/B}} \times \frac{CycleTime}{GateTime}, \tag{7}$$

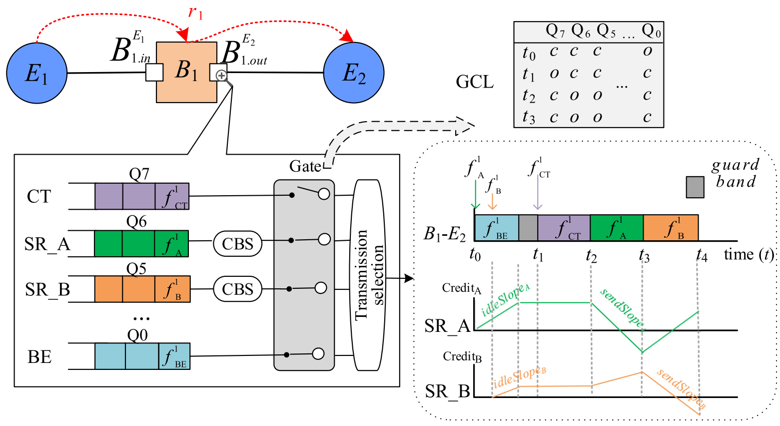

Fig. 5. Real-time scheduling modeling for TSN.

where $ActualBandwidth_{\mathrm{A/B}}$ is the actual bandwidth currently used by the queues related to traffic (SR_A or SR_B) and is unchanged; *CycleTime* is the gating cycle time; *GateTime* is the total time that the gate of the queue related to traffic (SR_A or SR_B) is open during the gating cycle; therefore, the rules for changing the credit value are also increased: when the queue door of AVB traffic is closed, the credit value remains unchanged until the queue gate is reopened.

The detailed transmission process of messages frames (f_{CT}^1, f_{A}^1, f_{B}^1 and f_{BE}^1) on link E_1-B_1 in TSN is shown in Figure 5; where Q7 represents the highest priority queue and Q0 represents the lowest priority queue; CT (purple part represents CT), AVB traffic, and BE traffic enter the corresponding priority queue according to PCP (e.g., CT enters Q7 queue); GCL represents the state of the gate (*o* or *c*) at time *t*, and we have set the gate of Q6 and Q5 (the queue where the AVB traffic is located) to remain open except during the guard band and CT stream transmission; the downward arrow represents the time when the frame arrives in the corresponding queue at $B_{1.out}^{E_2}$ and waits for the selected transmission; the gray square represents the guard band; the green and orange lines represent the credit value of SR_A and SR_B (i.e., $\mathrm{credit_A}$ and $\mathrm{credit_B}$), respectively.

(1) At time t_0, f_{BE}^1 is transmitting, f_{A}^1 and f_{B}^1 arrive at the queue in $\mathrm{B}_{1.out}^{E_2}$ successively, and the $\mathrm{credit_A}$ and $\mathrm{credit_B}$ begin to increase with $idleSlope_{\mathrm{A}}$ and $idleSlope_{\mathrm{B}}$, respectively.

(2) At time t_1, f_{BE}^1 transmission is completed, f_{CT}^1 arrives at the queue in $B_{1.out}^{E_2}$ and starts transmission, and a guard band mechanism is introduced, so that the gates of Q5 and Q6 are closed and the $\mathrm{credit_A}$ and $\mathrm{credit_B}$ remain unchanged.

(3) At time t_2, f_{CT}^1 transmission is completed, the gates of Q5 and Q6 are opened; f_{A}^1 and f_{B}^1 are waiting to be transmitted in the queues, and the $\mathrm{credit_A}$ and $\mathrm{credit_B}$ are greater than 0; f_{A}^1 is transmitted since f_{A}^1 has a higher priority than f_{B}^1; during f_{A}^1 transmission, the $\mathrm{credit_A}$ decreases with the $sendSlope_{\mathrm{A}}$ and the $\mathrm{credit_B}$ increases with the $idleSlope_{\mathrm{B}}$.

(4) At time t_3, f_{A}^1 transmission is completed, f_{B}^1 is waiting for transmission in the queue and the $\mathrm{credit_B}$ is greater than 0; at this moment, f_{A}^2 transmission begins; during f_{B}^1 transmission, the $\mathrm{credit_B}$ decreases with the $sendSlope_{\mathrm{B}}$ and the $\mathrm{credit_A}$ continues to increase with the $idleSlope_{\mathrm{A}}$.

In addition to the guard band mechanism, the frame preemption provided by 802.1Qbu and 802.1Br protocols also guarantees the real-time transmission of the CT stream while reducing the bandwidth waste caused by the guard band mechanism. The out-port that supports frame preemption mechanism can provide two MAC service interfaces: the **preemptible MAC (pMAC)** service interface and the **express MAC (eMAC)** service interface, as shown in Figure 6. When the

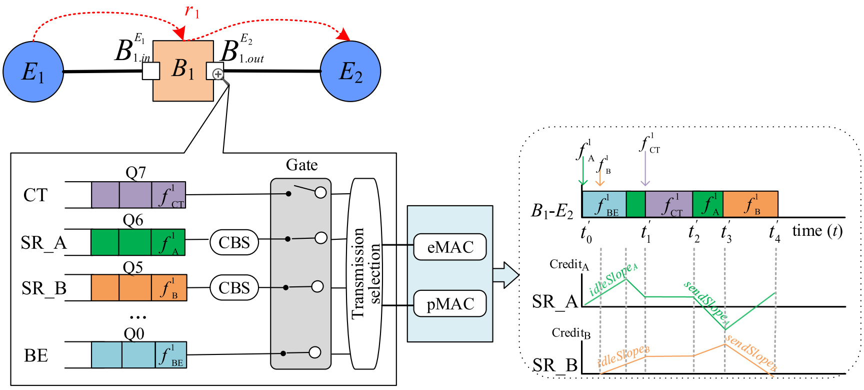

Fig. 6. Frame preemption for TSN.

frames are waiting to be transmitted in the queue, different status values are assigned to the frame according to the service interface used by the frame. The status value of the frame using the pMAC service interface is *preemptible*, which means that the frame can be preempted by the high-priority frame during transmission, and the transmission will continue after the transmission of the high-priority frame is completed; the status value of the frame using the eMAC service interface is *express*, which means that the frame can terminate the transmission of the *preemptible* frame. The same traffic class uses the same status value.

The detailed transmission process of the frame preemption mechanism is shown in Figure 6. CT frames are high-priority frames and use the pMAC service interface, so their status values are *express*; AVB frames and BE frames are low-priority frames and use the eMAC service interface, so their status values are *preemptible*. In other words, the mechanism terminates the transmission of low-priority frames (i.e., AVB and BE) when high-priority frames (i.e., CT) arrive; the remaining low-priority frames continue to be transmitted after the transmission of high-priority frames (i.e., CT) is completed, as shown Figure 6. When f^1_{CT} arrives at t'_1, the transmission of the f^1_{A} is terminated. When the transmission of f^1_{CT} is completed, the f^1_{A} resumes transmission. In addition, since the minimum transmission length of the frame is 64 bytes, the final transmission size of the frame cannot be less than 64 bytes. In other words, the frame cannot be preempted when the remaining part of the frame is less than 64 bytes.

2.5 Reliability Modeling for TSN

In order to improve the reliability of systems, TSN proposes fault-tolerance protocols (i.e., 802.1CB and 802.1Qca). The above protocols are combined to improve reliability by transmitting redundant replicas in network in parallel on disjoint paths; in other words, TSN mainly improves reliability through information redundancy (frame replicas) and link redundancy (multiple paths).

802.1CB provides the redundant mechanism of data transmission to solve the problems (e.g., frame loss and frame error) caused by network failures; the mechanism is mainly divided into two steps to achieve redundancy: (1) frame replication; (2) frame elimination.

(1) For frame replication, frames with redundant features must contain redundant tags, where an important field is **Sequence Number** (**SN**). Frame replication is realized by **Sequence Generation Function** (**SGF**), which generates duplicates with the same SN as the source frame. The source frames and duplicate frames are transmitted to destination end devices through different paths.

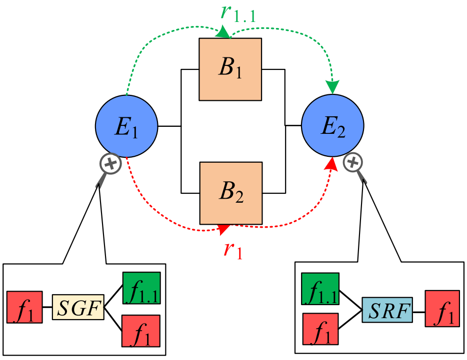

Fig. 7. Reliability modeling for TSN.

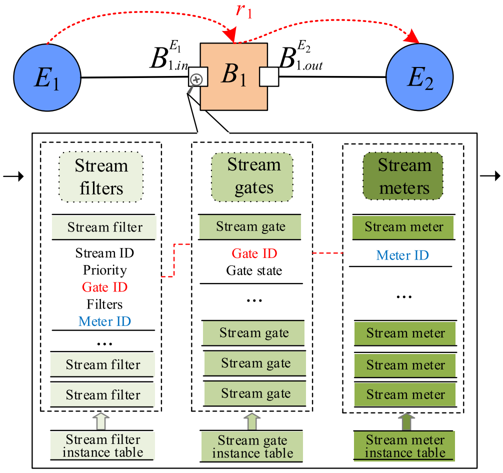

Fig. 8. Security modeling for TSN.

(2) For frame elimination, when duplicate frames with the same SN arrive at the destination end device, **Sequence Recovery Function** (**SRF**) eliminates duplicate frames by identifying frames with the same SN.

The redundancy process of the real-time data frame f_1 in TSN is shown in Figure 7, where frame f_1 is duplicated through SGF at E_1, and frame f_1 and $f_{1.1}$ have the same SN; frame f_1 and duplicate frame f_1 are transmitted to E_2 through r_1 and $r_{1.1}$, respectively; the frames f_1 and $f_{1.1}$ with the same SN are identified through SRF at E_2, thereby eliminating the duplicate frame $f_{1.1}$. If the frame f_1 is lost during transmission, the duplicate frame $f_{1.1}$ is transmitted to E_2, thereby ensuring the normal operation of system.

In Figure 7, redundant replicas are transmitted in network in parallel on disjoint paths; 802.1Qca calculates the data transmission path by combining the Intermediate Station to Intermediate Station (IS-IS) protocol and the **Shortest Path Routing** (**SPR**) method [86]. The frames in E_1 can be transmitted to E_2 through the path $r_{1.1}$ when a failure occurs in r_1, thereby realizing seamless redundancy and improving the reliability of the network.

2.6 Security Modeling for TSN

TSN proposes 802.1Qci to provide stream filtering and policing function for messages arriving at the in-port of bridge to prevent traffic overload, thereby improving network security. The process occurs after data stream arrives at the in-port of bridge and before it queues at the out-port.

The data processing process of stream filtering and policing function is mainly supported by the following components: (1) stream filter; (2) stream gate; and (3) stream meter. Each component has a corresponding parameter instance table, namely, stream filter instance table, stream gate instance table, and stream meter instance table, as shown in Figure 8. The stream filter instance table is an ordered list that consists of multiple stream filters. The stream entering the stream filter is assigned the stream gate and stream meter to achieve stream filtering and policing.

(1) For stream filter, the incoming frame is matched to the corresponding stream filter according to the *Stream Id* and *priority* parameter in stream filter, and frames that do not match the stream filter are discarded. *Zero or more filter specifications*, *Gate Id*, and *Meter Id* parameters in stream filter determine the filter specifications, stream gate, and stream meter related to the stream filter.

The filter specifications indicate that if the stream entering the stream filter exceeds the defined **Service Data Unit** (**SDU**), it is discarded.

(2) For stream gate, the filtered frames are transmitted in order according to the state of gate. The state of gate includes open and closed. The frame is allowed to pass when the gate is open, and the frame is not allowed to pass when the door is closed. If the frame is not transmitted on time in the correct time window, the frame will be discarded when the door is closed.

(3) For stream meter, the stream passing through the gate is policed by the meter specified by the stream filter. Streams using the same meter are determined by the stream classification rules. The stream classification rules identify stream set with the same forwarding and metering process based on the parameters of the frame (e.g., Destination MAC address, Source MAC address, Priority, and so on). The streams in the stream set obtained using the same classification rule use the same stream meter, in other words, the related bandwidth parameters of the streams in the same stream set are the same.

The stream meter instance contains parameters such as *Committed information rate*, *Committed burst size*, *Excess information rate*, *Excess burst size*, and so on. These parameters that depend on *Bandwidth Profile* set the bandwidth and burst size of the stream to police the data streams. The streams that do not meet these bandwidth allocations and traffic sizes are discarded. Therefore, 802.1Qci can solve network attacks and traffic overload problems. For example, stream filtering and policing functions will control the data stream by limiting the transmission bandwidth to improve the security of network when a DDoS network attack occurs in the network.

2.7 Applications from AVB to TSN

According to AVB and TSN protocols and features, as well as related analysis models, AVB and TSN can be applied to different fields.

(1) Applications of AVB. AVB belongs to the soft real-time network, so it is mainly used for the transmission of audio and video streams with real-time requirement in real-time critical systems [13]. AVB guarantees real-time data through time synchronization and forwarding mechanism, making it suitable for automotive systems. Avnu alliance [115] is an alliance organization dedicated to promoting in-vehicle Ethernet and AVB technology, and it has established a set of consistency testing solutions to achieve compatibility and interoperability among all audio and video devices in AVB network. Refs. [2] and [18] discussed the applicability of AVB in ADAS and automotive cameras, respectively. Refs. [61, 67] discussed the time synchronization mechanism in automotive networks. Ref. [67] studied the AVB synchronization mechanism by combining the FlexRay synchronization mechanism to ensure synchronization performance in automotive heterogeneous networks. Ref. [61] analyzed the applicability of AVB in automotive networks by measuring the accuracy of time synchronization mechanism under the influence of different environments (temperatures). Refs. [46, 82, 98, 113] discussed the forwarding mechanism in automotive networks. Ref. [82] configured CBS in AVB network to meet the real-time communication requirement of automotive audio and video applications. Ref. [46] considered the message forwarding between CAN and AVB in automotive systems. Ref. [98] studied the functions of stream reservation, CBS, and queue forwarding provided by AVB to meet the real-time requirement of automobiles, and are suitable for automotive **Electrical/Electronic** (**E/E**) architectures. Ref. [113] analyzed the end-to-end delay for AVB traffic in the automotive infotainment system through simulation; the experiment shows that AVB can realize the real-time requirement of in-vehicle communication.

AVB can be suitable for the avionics industry due to its real-time performance [39, 44, 106]. Ref. [106] depicted that the low-delay function of AVB can meet the real-time requirement of avionics digital audio. Ref. [44] studied the applicability of AVB in aviation networks by applying

AVB to the interphone equipment. Ref. [39] analyzed the end-to-end delay of traffic in avionics scenario by using CBS scheduling method.

AVB protocols meet the real-time requirement, but cannot guarantee network reliability and security when the network has a malicious fault.

(2) Applications of TSN. The emergence of TSN achieves hard real-time, high reliability, and security communication requirements for real-time critical systems (e.g., automobiles, industrial automation systems, and so on).

Many researchers have studied the applicability of TSN in automotive systems. Refs. [103, 105, 107] analyzed the performance of TSN protocols in automotive networks to meet the QoS for automotive communications. Ref. [40] introduced TSN into the automotive network, where a controller method that combines software and hardware is proposed to meet the real-time requirement in the new automobiles. Ref. [32] designed a modeling method based on **Logic Programming** (**LP**), which solved the problem of network configuration and verification when adding new sensors to automotive networks. Ref. [15] introduced the applicability of TSN in the automotive E/E Architecture to realize the security of the automotive network. Ref. [65] developed a TSN software integrated simulator by integrating TSN protocols (e.g., 802.1AS, Qbv, Qcc, and so on) to verify the applicability of TSN in autonomous driving. Ref. [42] proposed an automotive Ethernet network architecture based on **Software-Defined Networking** (**SDN**); it supports TSN real-time scheduling to achieve hard real-time requirement for communication. Refs. [5, 45, 68, 75] evaluated the applicability of TSN protocols in automobiles by using simulation. Refs. [68] and [5] studied the performance of IEEE 802.1Qch (Cyclic Queuing and Forwarding protocol) and IEEE 802.1Qcu (Frame preemption) in ADAS and infotainment systems through OMNeT++ simulation, respectively. Refs. [45, 75] evaluated the real-time performance of TSN in automobiles through OMNeT++ simulations.

TSN can be used in industrial fields with real-time, reliability, and security requirements [38, 120]. Ref. [111] studied the applicability of TSN in the field of Industry 4.0 by analyzing the working principle of TSN protocols; the research shows that TSN can meet the real-time and flexibility requirements for industry. Ref. [70] studied the real-time performance provided by IEEE 802.1Qbv protocol of TSN for industrial automation communication systems. Ref. [110] proposed an improved PTP that provides accurate time synchronization of network nodes, which makes TSN applicable to the wireless sensor deployed in critical control and automation applications. Ref. [101] proposed a high-precision synchronization method with integrating software and hardware through extending TSN to wireless networks, thereby realizing time synchronization for TSN-based hybrid networks. Ref. [33] introduced TSN to wireless communication systems and established a TSN toolbox with message processing and management to ensure the real-time and reliable functions of wireless communication systems. Ref. [22] designed a hybrid network with wired network and wireless protocols collaborative design; the hybrid network combines TSN and TDMA-based MAC protocol to achieve real-time and deterministic communication in industrial control systems. In addition, fog computing [97], machine learning [80], 5G [41], and mobile edge computing [128] combined with TSN can be applied in automotive, industrial automation, and control systems.

3 END-TO-END DELAY ANALYSIS FROM AVB TO TSN

AVB and TSN messages with real-time requirement must meet the end-to-end delay to ensure the safe operation of system, that is, the end-to-end delay of message does not exceed its deadline. The end-to-end of the message during transmission is mainly affected by shaping algorithms and different types of traffic interference (i.e., low-priority traffic interference, peer-priority traffic interference, and high-priority traffic interference). The shaping algorithms control the

Table 4. End-to-End Delay Analysis for AVB

Reference	For AVB/TSN	Method	Factors considered	Feature of method
[60]	AVB	CBS	CBS	It calculates the WCD of SR traffic.
[91]	AVB	CBS	CBS, low-priority traffic interference	It further considers the WCD of SR traffic under the interference of low priority traffic.
[43]	AVB	Network calculus	CBS	It uses bandwidth allocation strategy to ensure the end-to-end delay of SR traffic.
[95]	AVB	Network calculus	CBS	It uses network calculus calculation to calculate the WCD of SR traffic.
[23]	AVB	Formal analysis method	low-priority traffic interference	It uses CPA to calculate the WCD bounds of SR traffic in the network.
[25]	AVB	Formal analysis method	low-priority traffic interference	It analyzes the end-to-end delay of traffic for different network topologies based on the CPA.
[14]	AVB	Formal analysis method	CBS, low-priority and high-priority traffic interference	It analyzes the impact of traffic shaper on high priority messages and reduced the pessimism of WCD analysis.
[16]	AVB	Formal analysis method	the high-priority and low-priority traffic interference	It provides the WCD for medium-priority traffic by defining the eligible interval.
[17]	AVB	Formal analysis method	CBS, the high-priority traffic interference	It obtains the strict upper limit of WCD for medium-priority traffic by analyzing the impact of CBS on high-priority traffic.
[69]	AVB	Trajectory approach	CBS	It analyzes the WCD of SR_A, SR_B, and BE traffic based on serialization constraints of frame transmission.

deterministic transmission of traffic, and the resource competition among different types of traffic leads to message delay. At present, the main methods for end-to-end delay analysis include network calculus, formal analysis method, and trajectory method, and so on.

3.1 End-to-End Delay Analysis for AVB

In this article, we review the end-to-end delay analysis methods for AVB, which are mainly divided into: (1) network calculus; (2) formal methods; and (3) trajectory approach, as shown in Table 4.

AVB traffic is shaped through CBS to ensure transmission delay in AVB network. Refs. [60, 91] used CBS to calculate the WCD of SR traffic; where Ref. [91] further considered the WCD of SR traffic under the interference of low-priority traffic. Based on the impact of CBS on traffic transmission, Refs. [43, 95] calculated the end-to-end delay of traffic by using network calculus. Ref. [43] analyzed the impact of CBS on traffic (i.e., SR_A, SR_B, and BE) using network calculus and established a shaping model; this model uses a bandwidth allocation strategy to solve the link competition among various types of traffic during transmission, thereby reserving more bandwidth for BE traffic while ensuring the end-to-end delay of SR traffic in AVB network. Ref. [95] studied the forwarding mechanism of 802.1Qav and used network calculus method to calculate the WCD of SR traffic.

The formal method is another main method to analyze the end-to-end delay analysis for AVB. Ref. [23] proposed the WCD analysis model to calculate the WCD bounds of SR traffic in the network while considering the interference of low-priority traffic; the core idea of the model is as follows: (1) the frame transmission problem in AVB network is mapped to the task scheduling problem of the **Compositional Performance Analysis** (**CPA**) system; (2) a formal method (CPA) based on the busy period is proposed to calculate the WCD of the SR traffic in the network. Ref. [23] focused on model conversion, and Ref. [25] focused on the formal analysis of WCD and further analyzes the end-to-end delay of traffic for different network topologies (i.e., star topology, linear topology, and clustered linear topology) based on the WCD analysis model in Ref.[23].

Considering that Refs. [23, 25] calculated WCD under the interference of lower priority traffic and the calculation results based on busy period are pessimistic. Refs. [14, 16, 17] improved it to reduce the pessimism based on busy period analysis method. The specific details of the improvement are as follows: (1) Ref. [14] proposed an improved WCD method for SR_A traffic while considering the interference of low-priority and peer-priority traffic; the method derives the WCD calculation formula of SR_A traffic by analyzing the blocking effect of CBS on message transmission; meanwhile, the WCD calculation formula for SR_B traffic is proposed while considering the interference of high-priority traffic; (2) Ref. [16] proposed the formal analysis method by defining the eligible interval instead of busy period to provide the WCD for medium-priority traffic (i.e., SR_B) while considering the interference of high-priority (i.e., SR_A) and low-priority traffic (i.e., BE); and (3) Ref. [17] improved the WCD analysis method in Ref.[16] by analyzing the impact of CBS on the credit value of high-priority traffic, thereby obtaining the strict upper limit of WCD for medium-priority traffic. Compared with Ref. [16], the WCD analysis method proposed by Ref. [17] has lower complexity and provides tighter WCD results. Ref. [69] described an improved trajectory method for AVB networks to analyze the WCD of various types of traffic (i.e., SR_A, SR_B, and BE); the method considers the impact of CBS on various types of traffic and the transmission sequence of constrained frames.

3.2 End-to-End Delay Analysis for TSN

Similar to the end-to-end delay analysis methods for AVB, the formal methods and network calculus are also the main methods of the end-to-end delay analysis for TSN. Table 5 lists the end-to-end delay analysis methods for TSN considering different influencing factors.

In TSN, TAS is proposed to achieve the hard real-time transmission of critical streams by controlling GCL, and other shaper technologies are gradually developed for data streams transmission [85, 116, 117]. Ref. [117] conducted formal timing analysis on TAS, **Peristaltic Shaper** (**PS**), and **Burst-Limiting Shaper** (**BLS**) to calculate the WCD range of critical streams while considering the interference of all priority traffic (i.e., low-priority traffic, peer-priority traffic, and high-priority traffic interference); experiments show that TAS can achieve the best performance of end-to-end delay while ensuring network node time synchronization. Ref. [116] performed formal timing analysis on BLS to calculate the WCD range of critical streams while considering the interference of all priority traffic (i.e., low-priority traffic, peer-priority traffic, and high-priority traffic interference). Ref. [85] studied the emerging **Asynchronous Traffic Shaper** (**ATS**) and TAS through formal analysis methods; experiments show that ATS can ensure the deterministic transmission of real-time streams, but compared with TAS, it is not suitable for deterministic transmission of periodic critical traffic because it provides fairness for all traffic. Ref. [81] analyzed the delay of AVB frames under integrated scheduling based on CBS and TAS while considering the interference of all priority traffic (i.e., low-priority traffic, peer-priority traffic, and high-priority traffic interference) in TSN. Ref. [112] proposed a formal method to calculate the safety upper bound of non-critical streams; the method analyzes the interference of critical streams (i.e., high-priority traffic) to non-critical streams in TSN. Ref. [117] proposed the CPA framework to analyze the WCD of critical streams under the peer-priority interference; where the critical streams have exclusive communication links during transmission by setting GCL to control the transmission windows not to overlap, so that it is not affected by the non-critical streams.

Table 5. End-to-End Delay Analysis for TSN

Reference	**For AVB/TSN**	**Method**	**Factors considered**	**Feature of method**
[117]	TSN	Formal analysis method	TAS, PS, all priority traffic interference	It calculates the WCD range of critical streams while considering the interference of all priority traffic.
[116]	TSN	Formal analysis method	BLS, all priority traffic interference	It calculates the WCD range of critical streams while considering the interference of all priority traffic.
[85]	TSN	Formal analysis method	ATS and TAS	Compared with TAS, ATS is not suitable for deterministic transmission of periodic critical traffic because it provides fairness for all traffic.
[81]	TSN	Formal analysis method	TAS, CBS, all priority traffic interference	It analyzes the delay of AVB frames under hierarchical scheduling based on CBS and TAS.
[112]	TSN	Formal analysis method	high-priority traffic interference	It calculates the safety upper bound of non-critical streams under the interference of critical streams.
[124]	TSN	Network calculus	TAS	It analyzes the WCD of critical streams.
[127]	TSN	Network calculus	CBS	It sets the parameters $idleSlope_{\mathrm{A/B}}$ and $sendSlope_{\mathrm{A/B}}$ to ensure the upper bound of the WCD for AVB streams.
[125]	TSN	Network calculus	CBS	It introduces the shaping arrival curves.
[126]	TSN	Network calculus	CBS	It considers the credit value of AVB traffic during the guard band.

Considering the particularity of GCL setting in Refs. [117] and [124] considered GCL transmission windows in different overlapping situations and analyzed the WCD of critical streams using network calculus. With the development of real-time critical systems, a large number of devices connected to systems generate massive amounts of data with different priorities, mainly including CT, AVB traffic, and BE traffic. In the mixed criticality, the guard band mechanism and preemption mode are introduced [76, 87, 88] to ensure the end-to-end delay of critical streams. However, these mechanisms sacrifice the transmission of non-critical streams, resulting in the large WCD for non-critical streams, which may cause them to be unschedulable. Ref. [127] analyzed the WCD of AVB streams in TSN under the influence of the above mechanisms (i.e., guard band mechanism and the preemption mode) by using network calculus; the upper bound of the WCD for AVB streams is further ensured by setting the parameters (i.e., $idleSlope_{\mathrm{A/B}}$ and $sendSlope_{\mathrm{A/B}}$) to control the transmission of AVB streams. Refs. [125, 126] improved the WCD analysis method in [127] to make it suitable for different classes of AVB traffic (i.e., AVB traffic has any class: SR_A, SR_B, SR_C,...) while obtaining more strict WCD bound for AVB traffic: (1) Ref. [125] proposed an improved network calculus-based method by introducing the shaping arrival curves with combining the link transmission speed and CBS impact on AVB traffic; (2) Ref. [126] put forward an improved network calculus-based method, which considers the credit value of AVB traffic during the guard band.

4 REAL-TIME SCHEDULING FROM AVB TO TSN

In real-time critical systems (e.g., automobiles), there are different types of communication traffic. The data scheduling mechanism is the key to the reliable and timely transmission of messages. In recent years, researches on data scheduling have been a major concern of researchers.

Table 6. Real-Time Scheduling for AVB

Reference	For AVB/TSN	Research type	Method	Feature of method
[39]	AVB	Scheduling algorithm	CSB, SPQ, and FIFO	CBS solves SPQ starvation problem and reduces the delay of low priority traffic, and FIFO provides the worst delay.
[100]	AVB	Scheduling algorithm	CBS	CBS can schedule data frames in advance in the sudden situation.
[3]	AVB	Scheduling algorithm	TABS	It designs the idleSlope and sendSlope parameters to ensure low delay of ST.
[4]	AVB	Scheduling algorithm	The bandwidth over-reservation method	It increases the specified bandwidth of traffic type.
[63]	AVB	Scheduling algorithm	Bandwidth allocation strategy	It adjusts the MTU size of ST and AVB traffic, and found the optimal bandwidth allocation ratio of ST.
[73]	AVB	Scheduling algorithm	Improved AVB model	It combines with 802.1 Qav scheduling mechanism to achieve real-time transmission of time-triggered streams and event-triggered streams.
[102]	AVB	Scheduling algorithm	A shaper algorithm	It can be used to transmit time-triggered traffic and event-triggered traffic.

4.1 Real-Time Scheduling for AVB

In terms of AVB message scheduling, AVB proposes CBS to ensure the real-time transmission of audio and video data streams. Based on CBS, different scheduling methods are proposed to transmit network traffic, as shown in Table 6.

Ref. [39] compared the influence of scheduling methods (i.e., CSB, SPQ, and FIFO) on the WCD of scheduling streams by using OMNeT ++ simulation tool; the results show that compared to SPQ scheduling algorithm, CBS solves SPQ starvation problem (low-priority traffic is starving for a long time) by providing available transmission time slots for low-priority traffic, thereby reducing the delay of low-priority traffic; and FIFO provides the worst delay. Ref. [100] proved that the CBS algorithm in AVB network can schedule data frames in advance in a sudden situation. However, SR traffic class and CBS algorithm cannot be used for real-time critical control data streams transmission [91].

To transmit real-time critical control messages in AVB network, Ref. [3] introduced **Scheduled Traffic** (**ST**) in AVB network, unlike SR traffic, which is a high-priority traffic triggered and scheduled by time; where the **Time-Aware Blocking Shaper** (**TABS**) and the parameters (i.e., *idleSlope* and *sendSlope*) design method are proposed to ensure low delay of ST. However, the bandwidth allocation method in AVB network may cause traffic to be unschedulable due to the introduction of new traffic class. In order to solve the limitations of bandwidth allocation method in AVB network, Ref. [4] cited the concept of ST and named AVB network as AVB ST network; where a bandwidth over-reservation method is proposed to increase the specified bandwidth of traffic class in AVB ST network. Ref. [63] devised the bandwidth allocation strategy for ST by adjusting MTU size of ST and SR traffic to obtain the optimal bandwidth allocation ratio for ST. Ref. [73] constructed an improved AVB model to achieve low jitter requirement for in-vehicle network information while meeting the end-to-end delay requirement; this model adds a scheduling mechanism for time-triggered streams to AVB network, and combines with 802.1 Qav scheduling mechanism to achieve real-time transmission of time-triggered streams and event-triggered streams. Ref. [102] developed a shaper algorithm with combining CBS and time awareness to ensure the deterministic transmission of automotive network data; this algorithm can be used to transmit time-triggered traffic and event-triggered traffic.

Table 7. Real-Time Scheduling for TSN

Reference	For AVB/TSN	Research type	Method	Feature of method
[96]	TSN	Scheduling algorithm	GCL optimization strategies	It realizs the TT streams scheduling while ensuring that the number of queues is minimized.
[35]	TSN	Scheduling algorithm	GRAPS	It reduces the WCD of AVB streams while ensuring TT streams to be schedulable.
[21]	TSN	Scheduling algorithm	SMT and OMT	It configures key functional parameters that affect 802.1Qbv communication behavior.
[26]	TSN	Scheduling algorithm	Tabu search algorithm	It ensures effective scheduling while considering the waste of bandwidth caused by the guard bands.
[31]	TSN	Scheduling algorithm	Automated GCL synthesis model	It realizes the automation of GCL synthesis scheduling.
[64]	TSN	Routing optimization	GRAPS	It obtains the optimal routing of AVB streams and the smallest WCD.
[8]	TSN	Routing optimization	ILP	It combines redundancy and link capacity constraints to optimize the routing of TT streams and AVB streams.
[119]	TSN	Routing optimization	IACO	It considers optimizing the path length of routing.
[92]	TSN	Joint routing and scheduling	ILP	It determines the queue allocation of TT streams and GCL synthesis, and is suitable for finding the optimal solution for small cases.
[90]	TSN	Joint routing and scheduling	Genetic algorithm	It considers the interdependence of routing and scheduling, and combines scheduling constraints and routing constraints.
[36]	TSN	Joint routing and scheduling	Integration of KSP and GRASP	It considers the impact of GCL on AVB streams and optimizes the routing of AVB streams to meet the end-to-end delay.
[29]	TSN	Joint routing and scheduling	ILP	It is qualitatively affected by the size and structure of network and the transmission frequency of real-time transmission, and is suitable for small-scale networks.
[9]	TSN	Joint routing and scheduling	ILP scheduling and multipath routing algorithm	It achieve fault tolerance while meeting real-time requirement.
[123]	TSN	Joint routing and scheduling	A combined routing and scheduling algorithm	It meets the dynamic real-time requirements of control stream transmission when virtual machines migrate.
[122]	TSN	Joint routing and scheduling	Framework of joint scheduling and routing	It improves network schedulability by further considering the network topology optimization.

4.2 Real-Time Scheduling for TSN

In order to meet the hard real-time transmission of CT (e.g., Time-Triggered traffic: TT), TAS assigns GCL for each queue of each output port to achieve data scheduling. Due to the large number of terminal systems, switches, and the links between terminal systems and switches, a large number of routes are generated to realize the real-time transmission of control streams. In other words, traffic is controlled and transmitted from the perspective of time and space. Therefore, this article analyzes the real-time transmission of network traffic from three aspects: (1) scheduling algorithms; (2) routing optimization; and (3) joint scheduling and routing, as shown in Table 7.

(1) In terms of scheduling algorithms for TSN: researchers have proposed different methods for analyzing and optimizing GCL in TSN, such as **Integer Linear Programming** (**ILP**), heuristic algorithms, **Satisfiability Modulo Theories** (**SMT**) and **Optimization Modulo Theories** (**OMT**), and so on.

Ref. [96] described the setting of GCLs as multi-objective combinatorial optimization problems and proposed GCL optimization methods including ILP, constructive heuristics, and meta-heuristics; these methods realize the scheduling of TT streams while ensuring that the number of queues is minimized. However, methods in Ref. [96] ignore the transmission of AVB streams, which will cause the large WCD of AVB streams and cannot be scheduled. Therefore, Ref. [35] proposed the GCL optimization method based on **Greedy Randomized Adaptive Search Procedures** (**GRAPS**) to reduce the WCD of AVB streams while ensuring TT streams to be schedulable. Ref. [21] determined the queues of frame allocation by using SMT and OMT solvers to meet the schedulability constraints. Ref. [26] studied no-wait packet scheduling problem in TSN by mapping it to No-Waiting for Job Shop Scheduling Problem in operations research; where the Tabu search algorithm is proposed to implement time sensitive streams scheduling while considering the waste of bandwidth caused by the guard bands.

Although TSN provides the 802.1Qbv protocol to manually set the network schedule, it is impractical for the network with the complex industrial scale and the large amount of data traffic. Therefore, it is essential to set up a network schedule that changes dynamically according to network requirements and response time, so as to meet the real-time transmission of data traffic. Ref. [31] proposed an automated strategy for GCL synthesis scheduling to actively configure feasible GCL synthesis: (1) the graphical model is proposed to create a network model that specifies the relevant data stream characteristics and constraints for the specific application; (2) the LP is used to generate scheduling constraints conditions for network model; and (3) the feasible GCL synthesis is obtained by using SMT solver to solve the constraint conditions.

(2) In terms of optimizing routing: Ref. [64] described AVB streams routing optimization strategy to make all frames schedulable and to minimize the WCD of AVB streams in TSN: (1) the **K-Shortest Path** (**KSP**) algorithm is used to reduce the search space; (2) the Cost Function is proposed to evaluate each solution; and (3) a heuristic algorithm based on GRASP is proposed to obtain the most optimal solution (i.e., the optimal routing of AVB streams and the smallest WCD). Ref. [8] proposed ILP-Based Multipath Routing algorithm to reduce the WCD of AVB streams; the algorithm combines redundancy features and link capacity constraints while optimizing both the TT streams routing and the AVB streams routing. Ref. [119] aimed at minimizing the path length of routing, and proposed an **improved ant colony optimization** (**IACO**) to achieve real-time scheduling of TT streams.

(3) In terms of joint scheduling and routing: the above methods optimize scheduling or routing independently without considering the interaction between scheduling and routing. We survey the advances of this issue. Ref. [92] realized that all frames in network are schedulable by considering three aspects (i.e., TT streams scheduling, the routing of TT streams, and the routing of AVB streams): (1) Dijkstra's algorithm is used to determine the TT streams routing; (2) an ILP algorithm for TT streams scheduling is proposed to determine the queue allocation of TT streams in the switch; meanwhile, the GCL synthesis (time-list) is proposed to control the opening and closing of queue gates; and (3) the GRAPS method [64] is used to optimize the routing of AVB streams. Ref. [90] proposed the heuristic scheduling method based on the genetic algorithm by considering the interdependence between routing and scheduling; the method improves the transmission efficiency and queue utilization of TT streams by combining scheduling constraints and routing constraints. However, Ref. [92] focused on the real-time transmission of TT streams and did not discuss the transmission of other traffic types in TSN. Ref. [36] studied this problem by integrating the heuristic algorithm (based on KSP [47]) and the meta-heuristic algorithm (GRASP), that is, a joint scheduling and routing algorithm; the joint algorithm considers the impact of GCL on AVB

Table 8. Reliability-Aware Design Methods for TSN

Reference	Method	Redundancy type	Failures	Feature of method
[121]	Warshall algorithm	information and link redundancy	node and link failures	It calculates the failure rate of network topology.
[94]	802.1 CB	information redundancy	information failures	It analyzes the seamless redundancy in TSN when the frame is loss.
[93]	Extended redundancy framework	information redundancy	information failures	It integrates 802.1 CB to ensure communication redundancy among real-time applications and systems.
[78]	Time redundancy strategy	time redundancy	transient failures	It replicates frames and retransmits in time.
[6]	Heuristic strategy and ILP	link and time redundancy	transient failures	It combines the network topology, routing, and scheduling to meet the redundant constraints for TT streams.
[7]	Time redundancy strategy	information and time redundancy	transient failures	It can effectively improve the network reliability.
[77]	Replicating the bridge	link redundancy	permanent failures	It replicates each node and CNCe connected to the bridge.
[37]	TRH, GRASP and TRO	link redundancy	link failures	These methods can reduce the cost of redundancy, especially GRASP, suitable for different network topologies.

streams transmission and optimizes AVB streams transmission to meet the end-to-end delay; the experimental evaluation of the algorithm is suitable for large-scale networks. Ref. [29] designed a joint scheduling and routing algorithm based on ILP; this algorithm sets the queue to be empty when the real-time stream is transmitted at the ingress port, thereby minimizing the end-to-end delay of real-time data streams. Ref. [9] combined ILP scheduling algorithm and multipath routing algorithm to achieve fault tolerance while meeting real-time requirement for TSN. Ref. [123] used a combined routing and scheduling algorithm (minimal distance tree construction-heuristic breadth first search: MDTC-HB) to meet the dynamic real-time requirement of hard real-time streams transmission in TSN when virtual machines migrate. Ref. [122] constructed a framework of joint scheduling and routing to find effective synthesis schedules; the framework improves the schedulability of the network by further optimizing the network topology.

5 RELIABILITY-AWARE DESIGN FOR TSN

Besides real-time requirement, reliability requirement is also important for real-time systems. According to the reliability model summarized in Section 2, we survey the recent advances of TSN reliability in terms of information redundancy and link redundancy. In addition, this article also discusses time redundancy methods. Table 8 lists the recent advances of reliability-aware design methods for different failures in TSN.

Network failures may occur on any link or node, which may cause frame loss. Therefore, Ref. [121] divided network failures into node failures and link failures. It studied the method of network failure rate for network designers to improve the reliability of the network; where the enumeration method is proposed to list all possible network failures, but this method becomes inefficient with the number of network nodes increasing; therefore, the Warshall algorithm is proposed to calculate the frame failure rate (including node failure rate and link failure rate) among any nodes in network. Network designers can design reliable network based on failure rate algorithm in Ref. [121], thereby reducing design cost. Ref. [94] analyzed the seamless redundancy in TSN by studying the working principle of 802.1 CB (i.e., frame replication and elimination) when the frame is loss. Ref. [93] proposed an extended redundancy framework for real-time applications and systems to achieve redundant communication among them; the framework integrates 802.1CB protocol and configures the redundancy mechanism through *Redundancy Profile*.

Network failures can be divided into transient failures and permanent failures from time perspective. For transient failures, due to the randomness and uncertainty of the time and location it generates, fault tolerance strategies of information redundancy and time redundancy are generally adopted. Ref. [78] adopted a time redundancy strategy by replicating frames and retransmitting in time to tolerate the transient failures in the network. Ref. [6] studied the seamless redundancy of TT streams in TSN by proposing a heuristic strategy; the heuristic strategy combines the network topology, routing, and scheduling to meet the redundant constraints for seamless redundant transmission of TT streams, thereby ensuring the real-time and reliability of TT streams transmission. Ref. [7] employed the time redundancy strategy to improve network reliability: (1) an ILP-based algorithm is proposed to determine the routing and replica of each AVB stream; the algorithm combines routing constraints, redundancy constraints, and link available capacity constraints; (2) the Mean Time To Detected Error is adopted to evaluate the reliability of the network; experiments show that the time redundancy strategy can effectively improve the network reliability. However, the replicas calculated by Ref. [7] are transmitted by the same link. If the link has a permanent failure, the system will be unavailable. Ref. [77] constructed a highly reliable architecture for TSN, which solves the problem of permanent failures by replicating the bridge; this architecture replicates each node and the **centralized network configuration element** (**CNCe**) connected to the bridge.

However, the redundancy increases the cost for real-time safety critical systems. Ref. [37] applied three methods (i.e., Topology and Routing Heuristics: TRH, GRASP, **Topology and Routing Optimization** (**TRO**)) to reduce the cost of network architecture while meeting the real-time and reliability requirements of real-time applications; these methods optimize network routing by integrating the network topology and stream routing, thereby solving the cost problem; experimental results show that these methods can significantly reduce the cost of network (especially TRO and GRASP) and obtain redundant disjoint paths to meet fault tolerance; however, TRO only finds optimal solutions for smaller network topologies, and it does not adapt well to the size of network topology; GRASP can effectively search the solution space and obtain the optimal solution in a reasonable time, and it is suitable for more different network scales.

6 SECURITY-AWARE DESIGN FOR TSN

With the development of real-time critical systems such as intelligent automobiles, industrial automation, and industrial control, network security issue has become increasingly prominent. In order to ensure the security of network, TSN defines the 802.1Qci protocol to block malicious devices or attacks like DDoS. At present, there are few researches on the security of TSN, but there are still different researchers who have discussed the enhancement of protocols, algorithms, and encryption mechanisms to ensure the security of network [34], as shown in Table 9.

Refs. [89, 118] established a simulation framework to evaluate the performance of the protocol for traffic filtering (i.e., 802.1Qci), where Ref. [118] proposed the scheduling method with combining 802.1Qci to realize data scheduling while considering data security. Ref. [11] adopted a security strategy management policy to ensure the security of communication data streams; the policy combines the security policy of each stream and the network security policy through 802.1Qci. Ref. [79] used a comprehensive method combining routing and scheduling to ensure the stability of

Table 9. Security-Aware Design Methods for TSN

Reference	Method	Feature of method
[89]	The simulation framework	It evaluates the performance of 802.1Qci protocol for traffic filtering.
[118]	The scheduling method with combining 802.1Qci	It ensures data security by combining 802.1Qci.
[11]	802.1Qci	It integrates the security policy of each stream with the network-wide security policy, and then deployed and managed the policy on switch.
[79]	Block encryption algorithms, binary search, and genetic algorithms	It ensures the security of applications.
[72]	Key management, frame replication and elimination, and VLAN segmentation	It combines safety and security.
[108]	The centralized configuration network mechanism	It combines TSN streams configuration with SGT.

applications in TSN; meanwhile, block encryption algorithm, binary search, and genetic algorithm are used to guarantee the security of applications.

At the beginning of automotive network design, safety and security should be considered at the same time to ensure the reliability and security of network. Ref. [72] discussed the interaction between safety and security in automotive network from three aspects (i.e., key management, frame replication and elimination, and **virtual local area network** (**VLAN**) segmentation); experiments show that cyber security is very important for system safety, and security must be considered when considering safety. Based on the centralized configuration network mechanism, Ref. [108] combined TSN streams configuration with **Secure Group Tags** (**SGT**) to achieve end-to-end security in TSN.

7 TSN IN AUTOMOTIVE DOMAIN

The real-time, reliability, and security features of TSN make it suitable for the automotive domain, where the automobile is an important, potential, and economically valuable application scenario for TSN. Therefore, the following sections discuss the application of TSN in automotive domain.

7.1 TSN-Based Automotive E/E Architecture

With the development of automobile intelligent driving, the data traffic and the number of ECUs continue to increase, thereby putting forward high-bandwidth and low-delay requirements for automotive E/E architecture. In order to meet the above requirements, the automotive E/E architecture has gradually shifted from integrated architecture to Domain-based backbone Ethernet automotive E/E architecture and Zone-based backbone Ethernet automotive E/E architecture [99].

(1) Domain-based TSN automotive E/E architecture. The Domain-based backbone Ethernet automotive E/E architecture uses Ethernet as the backbone network and integrates five domains of powertrain, chassis, ADAS, body, and infotainment; each domain has its own **Domain Control Unit** (**DCU**). At present, the new automotive E/E architecture considers TSN as the backbone Ethernet, as shown in Figure 9. The use of Domain-based TSN automotive E/E architecture improves the network bandwidth and computing power while maintaining the domain division characteristics of the existing distributed integrated architecture, thereby meeting the development requirements of intelligent automobiles. Therefore, the Domain-based TSN automotive E/E architecture is the preferred architecture for the mass production of intelligent automobiles by various automakers.

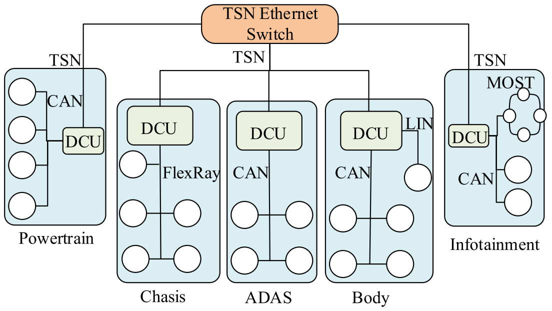

Fig. 9. Domain-based TSN automotive E/E architecture.

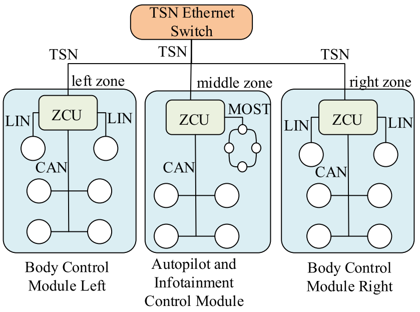

Fig. 10. Zone-based TSN automotive E/E architecture.

(2) Zone-based TSN automotive E/E architecture. In addition, the Zone-based backbone Ethernet automotive E/E architecture is the trend of a new generation of intelligent automotive E/E architecture. This architecture uses TSN as the backbone network and contains a hybrid topology of multiple network subsystems. Different from the Domain-based TSN automotive E/E architecture, Zone-based TSN automotive E/E architecture divides the “Zone” according to physical space instead of function. For example, Tesla Model 3 adopts the left zone, middle zone, and right zone; each zone has its own **Zone Control Unit** (**ZCU**), as shown in Figure 10. The architecture can directly cross-domain control (i.e., multi-domain control). It can arrange different functional electrical appliances with high-computing requirements in a zone, and meet the high-requirements of the whole automotive computing power by improving the computing power of the controller for a single zone.

7.2 End-to-End Delay Analysis for TSN-based Automotive E/E Architecture

Domain-based and Zone-based TSN automotive E/E architectures face the same limitations of end-to-end delay analysis.

(1) The pessimism of end-to-end delay analysis. The existing single-level network architecture of ECU-Gateway has evolved into the two-level network architecture of ECU-Domain/Zone-Central Controller. The change of network architecture makes the end-to-end delay analysis from the original two segments (i.e., source network domain to central gateway, central gateway to destination network domain) to at least four segments (i.e., source network domain to source domain controller, source domain controller to central controller, central controller to destination domain controller, destination domain controller to destination network domain), as shown in Figure 11. It is feasible to use the current piecewise accumulation methods (e.g., CPA, network calculus) to calculate the delay for the original two segments, because it has a low-impact on the overall pessimism of analysis. However, the current methods will produce too pessimistic analysis results for at least four segments. This overly pessimistic result will bring high-costs for the design of the network architecture.

(2) The delay caused by the cyber security strategies. With the development of IT and the increase of network wireless interfaces for intelligent internet automotive terminals, the risk of attacks on automotive networks is increasing. In the TSN-based automotive E/E architecture, the domain controller guarantees the internal communication security of subsystem through the cyber security strategies (e.g., security authentication, encryption, and decryption, and so on). Therefore, the end-to-end delay analysis for TSN-based automotive E/E architecture needs to further consider the delay caused by the cyber security strategies. However, the existing end-to-end analysis

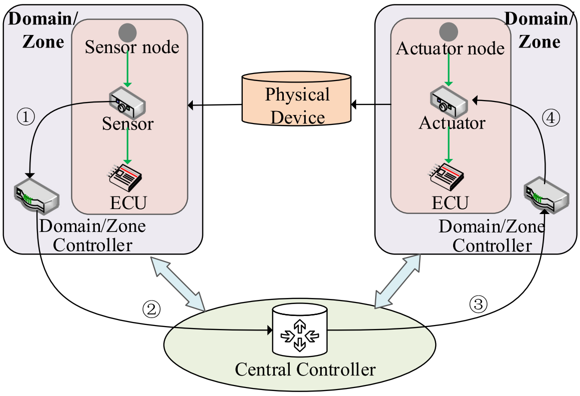

Fig. 11. End-to-end delay analysis for TSN-based automotive E/E architecture.

methods cannot calculate the delay caused by the cyber security strategies. In addition, functional safety and cyber security affect each other in automotive networks, and the end-to-end delay analysis method that considers functional safety and cyber security has become a new trend.

For the above limitations of end-to-end delay analysis, it is a research trend to propose a holistic end-to-end delay analysis method to analyze all segments. The method should consider the delay influencing factors of the message, including the internal delay of the bridge, the interaction among segments, and the impact of security strategies; where the internal delay of the bridge is caused by traffic shaping mechanisms (e.g., TAS, CBS) and the interference of different priority (i.e., high-priority, peer priority, and low-priority) traffic. Then, this method can obtain the tight delay upper bound and reduce pessimism by adjusting the delay influencing factors of the message in the process of cross-domain transmission.

7.3 Real-Time Scheduling for TSN in Automobiles

There is data with different priorities in automotive networks, mainly including TT traffic, AVB traffic (i.e., SR_A stream and SR_B stream), and BE traffic. Although TSN provides the data scheduling mechanism to ensure the data real-time requirement, it cannot completely solve the problem of traffic scheduling in automotive networks.

(1) GCL configuration. The current real-time scheduling methods allocate a fixed highest priority queue for TT streams and implement deterministic transmission of TT streams by closing the gates of other non-TT stream queues, thereby making the message arrival time predictable and improving hard real-time performance. The queue allocation scheme is feasible for small-scale real-time networks, but it is unrealistic for automotive networks with a huge amount of TT streams transmission. If multiple queues are allocated for TT streams and GCL is used to control the open and close states of all stream queues, TT streams transmission may be delayed. The reason is that TT streams in multiple queues will inevitably compete for the same link at the same time, thereby making TT streams transmission uncertain. Therefore, it is important to design the TSN bridge that supports different priorities of TT stream classes (i.e., class T1, class T2, . . . , where T1 has the highest priority); this bridge can dynamically allocate transmission queues and configure GCL according to the data transmission requirement in networks, thereby ensuring the deterministic transmission of TT streams under multiple queues.

(2) Real-time scheduling of mixed traffic. In addition to guaranteeing the deterministic transmission of TT streams, the transmission of AVB streams cannot be ignored. The transmission of AVB streams is restricted by the credit value in CBS and the guard band (i.e., AVB streams can be transmitted when the credit value is non-negative and the gate is open). Although CBS can guarantee the alternate transmission of SR_A streams and SR_B streams to prevent SR_B streams starvation, it ignores the deadline of data streams, thereby resulting in data streams unschedulable; moreover, when the link is idle, AVB streams cannot be transmitted due to the negative credit value, thereby causing the waste of bandwidth. The guard band mechanism ensures that the link is idle before transmitting TT streams and does not allow new AVB streams to be transmitted. As the number of the guard band increases, AVB streams may be unschedulable and the waste of bandwidth increases. To improve the transmission of AVB streams and bandwidth utilization, it is necessary to propose the method to enhance the transmission of AVB streams; this method can monitor the deadline of AVB streams in time and dynamically change the credit value according to their deadline, thereby achieving real-time transmission of AVB streams.

7.4 P802.1DG—TSN Profile for Automobiles

At present, IEEE 802.1 Working Group has established P802.1DG (TSN Profile for Automotive In-Vehicle Ethernet Communications) [58] for the application of TSN in automobiles to meet the hard real-time, reliability, and security requirements of automotive communication. P802.1DG is an ongoing standardized profile specifically for TSN applying to the automotive domain and will cover the following aspects: (1) The introduction of automotive communication architecture, traffic classes (i.e., TT streams, AVB streams, and BE streams), queues, and so on; (2) The introduction of TSN tools and their performance comparison, such as CBS, TAS; (3) The functions, characteristics, principles, and configurations of TSN protocols; (4) The performance evaluation of TSN communication, such as hard real-time, reliability, and security; (5) Actual use cases and future considerations. The standardized P802.1DG profile provides the configuration guideline for network designers and implementers. According to the different requirements of automotive applications, the network designers and implementers can establish the automotive communication network by adopting relevant TSN functions, protocols, configurations, and so on. In addition, TSN is co-operating with IEEE 802.1 Security standards to provide the security configuration standard for P802DG, thereby enhancing the security of automotive communication.

8 FUTURE TRENDS OF TSN

Since TSN provides real-time, deterministic, reliable, and secure data transmission, it is currently an advanced communication technology being actively promoted by the automotive and industrial industries. The development of TSN will become a common concern topic of the automotive and industrial industries. The following sections discuss the future trends of TSN.

8.1 Automated Configuration of TSN

TSN provides the data scheduling mechanism to meet the real-time requirement of real-time critical systems. For CT with the hard real-time requirement, TSN provides the 802.1Qbv protocol to manually offline set GCL, thereby realizing real-time transmission of CT. It is feasible for the communication of small-scale networks to manually offline set GCL, but it is impractical for the network with the complex industrial scale and a large amount of data traffic. Therefore, the more flexible and easy-to-operate configuration methods for TSN need to be considered, especially for offline configuration of large-scale network communication. It is necessary for TSN to implement an automatic configuration mechanism for network scheduling according to network requirements and response time, so as to meet the real-time transmission of CT. For AVB streams (i.e., SR_A and SR_B) with the soft real-time requirement, their transmission is controlled by CBS. The CBS stipulates that SR_A and SR_B streams will only be transmitted when the credit value is non-negative, and the increase and decrease of the credit value for SR_A and SR_B streams depend on bandwidth guarantee parameters: $idleSlope_{\mathrm{A/B}}$ and $sendSlope_{\mathrm{A/B}}$. Therefore, it is important for improving the transmission of AVB streams to configure appropriate bandwidth guarantee parameters. In addition, TSN provides 802.1CB and 802.1Qca protocols (i.e., information redundancy and link redundancy) to enhance the reliability of communication. Therefore, how many duplicate messages are set and how to set the transmission path of these duplicate messages are critical to the reliability of communication.

8.2 Security Enhancement of TSN

TSN is the preferred advanced communication technology for real-time critical systems such as intelligent automobiles, aerospace, industrial control, and industrial automation. The data generated by devices in real-time critical systems is increasing, the interaction among devices is increasing, and the network interface is exposed to the public environment. These factors increase the chance of being attacked by attackers, thereby causing unpredictable risks to systems. Although TSN provides the 802.1Qci protocol to filter burst frames (e.g., traffic overload, DOS attacks, and so on), it still lacks protection mechanisms for cyber security (i.e., the confidentiality and integrity of data). At present, blockchain [83] provides encryption and decryption strategies to support the confidentiality of data, including symmetric encryption and asymmetric encryption. The two strategies are suitable for different requirements and can also be used in combination to form a hybrid encryption strategy. Blockchain not only provides the confidentiality strategy of data but also proposes the Hash algorithm to ensure the integrity of data. Therefore, the use of blockchain in automotive network TSN is a development trend to ensure data confidentiality and integrity. In addition, the IEEE 802.1 TSN Task Group is cooperating with IEEE 802.1 Security standards to improve communication security. Therefore, how to enhance the cyber security mechanism of TSN to ensure the confidentiality and integrity of data is a common concern topic of the automotive and industrial industries.

8.3 Congestion Control of TSN

The prerequisite for achieving communication requirements (i.e., hard real-time, reliability, and security) is that the automotive network has good congestion control capability. Congestion refers to the phenomenon that the number of data packets arriving in a certain part of the communication subnet is too much, making this part of the network unable to process these data packets in time. This phenomenon causes this part and even the entire network performance to decline, and even causes network communication services to stop.

At present, TSN focuses on the design of bridges and ignores the congestion control of the entire network. The deep learning model is the current popular network congestion control technology. Some advanced deep learning models have been used for network congestion control, such as Recurrent Neural Network, Deep Neural Network, and Dimensional Convolutional Neural Network [10]. Although deep learning models can effectively achieve congestion control, there is still a common challenge: the number of congested sample streams in the training phase is insufficient, and is far lower than the number of normal sample streams. The imbalance between the congested sample streams and the normal sample streams hinders the congestion control capability of these models. Therefore, how to generate the sufficient congested sample streams through the deep learning model to improve the congestion control capability of TSN is an issue worthy of study in the future.

9 CONCLUSION

This article provides the recent advances and future trends of modeling and design methodologies from AVB to TSN according to different communication requirements (i.e., hard real-time, reliability, and security) in real-time critical systems. This article surveys the modeling from AVB to TSN, including: (1) AVB and TSN system modeling; (2) end-to-end delay modeling; (3) real-time scheduling modeling; (4) reliability modeling; and (5) security modeling. Based on models, this paper surveys the recent research advance in the following aspects from AVB to TSN: (1) end-to-end delay analysis from AVB to TSN, mainly including formal methods and network calculus; (2) real-time scheduling from AVB to TSN, including scheduling algorithms, routing optimization, and joint scheduling and routing; (3) reliability-aware design for TSN, including information redundancy, link redundancy, and time redundancy; and (4) security-aware design for TSN. This paper future takes the automotive use case as an example to discuss the current status, challenges, and trends of TSN applying in automotive domain mainly including: (1) TSN-based automotive E/E architecture; (2) end-to-end delay analysis for TSN-based automotive E/E architecture; (3) real-time scheduling for TSN in automobiles; and (4) P802.1DG - TSN profile for automobiles. Finally, the paper discusses the future trends of TSN, including automated configuration of TSN, security enhancement of TSN, and congestion control of TSN.

The real-time modeling and analysis from AVB to TSN are carried out from two aspects: (1) end-to-end delay modeling and analysis methods provide the upper bound of data response; (2) real-time scheduling modeling and analysis methods ensure timely and reliable data transmission. Reliability modeling and design methods for TSN enhance fault-tolerant communication capabilities of network. Security modeling and design methods for TSN provide security protection strategies for communication systems. The above four aspects form the core of recent advances in real-time Ethernet modeling and design methodologies from AVB to TSN.

The main purpose of this article is to provide basic theoretical reference for researchers interested in modeling and design methodologies for AVB and TSN. We hope that more researchers can improve the TSN design method and apply it to different research fields.

ACKNOWLEDGMENTS

The authors express their gratitude to the associate editor and two anonymous reviewers whose constructive comments have helped to improve the manuscript.

REFERENCES

[1] 2018. IEEE Standard for Local and Metropolitan Area Network–Bridges and Bridged Networks. IEEE Std 802.1Q-2018 (Revision of IEEE Std 802.1Q-2014) (July 2018), 1–1993. https://doi.org/10.1109/IEEESTD.2018.8403927.

[2] G. Alderisi, G. Iannizzotto, and L. L. Bello. 2012. Towards IEEE 802.1 ethernet AVB for advanced driver assistance systems: A preliminary assessment. In *Proceedings of the 2012 IEEE 17th International Conference on Emerging Technologies Factory Automation.* 1–4.

[3] G. Alderisi, G. Patti, and L. L. Bello. 2013. Introducing support for scheduled traffic over IEEE audio video bridging networks. In *Proceedings of the 2013 IEEE 18th Conference on Emerging Technologies Factory Automation.* 1–9.

[4] Mohammad Ashjaei, Gaetano Patti, Moris Behnam, Thomas Nolte, Giuliana Alderisi, and Lucia Lo Bello. 2017. Schedulability analysis of ethernet audio video bridging networks with scheduled traffic support. *Real-Time Systems* 53, 4 (Feb. 2017), 526–577.

[5] Mohammad Ashjaei, Mikael Sjödin, and Saad Mubeen. 2021. A novel frame preemption model in TSN networks. *Journal of Systems Architecture* 116 (Jan. 2021), 102037.

[6] A. A. Atallah, G. B. Hamad, and O. A. Mohamed. 2018. Fault-resilient topology planning and traffic configuration for IEEE 802.1Qbv TSN networks. In *Proceedings of the 2018 IEEE 24th International Symposium on On-Line Testing and Robust System Design.* 151–156.

[7] Ayman A. Atallah, Ghaith Bany Hamad, and Otmane Ait Mohamed. 2018. Reliability-aware routing of avb streams in tsn networks. In *Proceedings of the International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems.* Springer, 697–708.

[8] Ayman A. Atallah, Ghaith Bany Hamad, and Otmane Ait Mohamed. 2019. Multipath routing of mixed-critical traffic in time sensitive networks. In *Proceedings of the International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems.* Springer, 504–515.

[9] A. A. Atallah, G. B. Hamad, and O. A. Mohamed. 2020. Routing and scheduling of time-triggered traffic in time-sensitive networks. *IEEE Transactions on Industrial Informatics* 16, 7 (Nov. 2020), 4525–4534.

[10] M. Bachl, T. Zseby, and J. Fabini. 2019. Rax: Deep reinforcement learning for congestion control. In *Proceedings of the 2019 IEEE International Conference on Communications*. 1–6.

[11] Robert Barton, Maik Seewald, and Jerome Henry. 2018. Management of IEEE 802.1Qci security policies for time sensitive networks (TSN). (Oct. 2018). Retrieved December 10, 2020 from https://www.tdcommons.org/dpubs_series/1541.

[12] L. L. Bello. 2014. Novel trends in automotive networks: A perspective on Ethernet and the IEEE Audio Video Bridging. In *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation*. 1–8.

[13] L. L. Bello, R. Mariani, S. Mubeen, and S. Saponara. 2019. Recent advances and trends in on-board embedded and networked automotive systems. *IEEE Transactions on Industrial Informatics* 15, 2 (Nov. 2019), 1038–1051.

[14] U. D. Bordoloi, A. Aminifar, P. Eles, and Z. Peng. 2014. Schedulability analysis of Ethernet AVB switches. In *Proceedings of the 2014 IEEE 20th International Conference on Embedded and Real-Time Computing Systems and Applications*. 1–10.

[15] S. Brunner, J. Roder, M. Kucera, and T. Waas. 2017. Automotive E/E-architecture enhancements by usage of ethernet TSN. In *Proceedings of the 2017 13th Workshop on Intelligent Solutions in Embedded Systems*. 9–13.

[16] J. Cao, P. J. L. Cuijpers, R. J. Bril, and J. J. Lukkien. 2016. Tight worst-case response-time analysis for ethernet AVB using eligible intervals. In *Proceedings of the 2016 IEEE World Conference on Factory Communication Systems*. 1–8.

[17] Jingyue Cao, P. J. L. Cuijpers, Reinder J. Bril, and Johan J. Lukkien. 2018. Independent WCRT analysis for individual priority classes in Ethernet AVB. *Real-Time Systems* 54, 4 (Oct. 2018), 861–911.

[18] B. M. Cheon, C. Yi, and J. W. Jeon. 2014. Hardware platform for automotive camera based on Ethernet AVB. In *Proceedings of the the 18th IEEE International Symposium on Consumer Electronics*. 1–2.

[19] S. Coicheci and I. Filip. 2020. Self-driving vehicles: Current status of development and technical challenges to overcome. In *Proceedings of the 2020 IEEE 14th International Symposium on Applied Computational Intelligence and Informatics*. 000255–000260.

[20] Silviu S. Craciunas, R. Serna Oliver, and TC AG. 2017. An overview of scheduling mechanisms for time-sensitive networks. In *Proceedings of the Real-Time Summer School LÉcole dÉté Temps Réel*. 1551–3203.

[21] Silviu S. Craciunas, Ramon Serna Oliver, Martin Chmelík, and Wilfried Steiner. 2016. Scheduling real-time communication in IEEE 802.1 Qbv time sensitive networks. In *Proceedings of the 24th International Conference on Real-Time Networks and Systems*. 183–192.

[22] C. Cruces, R. Torrego, A. Arriola, and I. Val. 2018. Deterministic hybrid architecture with time sensitive network and wireless capabilities. In *Proceedings of the 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation*. Vol, 1. 1119–1122.

[23] Jonas Diemer, Jonas Rox, and Rolf Ernst. 2012. Modeling of ethernet AVB networks for worst-case timing analysis. *IFAC Proceedings Volumes* 45, 2 (April 2012), 848–853.

[24] J. Diemer, J. Rox, R. Ernst, F. Chen, K. Kremer, and K. Richter. 2012. Exploring the worst-case timing of Ethernet AVB for industrial applications. In *Proceedings of the 38th Annual Conference on IEEE Industrial Electronics Society*. 3182–3187.

[25] J. Diemer, D. Thiele, and R. Ernst. 2012. Formal worst-case timing analysis of Ethernet topologies with strict-priority and AVB switching. In *Proceedings of the 7th IEEE International Symposium on Industrial Embedded Systems*. 1–10.

[26] Frank Dürr and Naresh Ganesh Nayak. 2016. No-wait packet scheduling for IEEE time-sensitive networks (TSN). In *Proceedings of the 24th International Conference on Real-Time Networks and Systems*. 203–212.

[27] K. Erwinski, M. Paprocki, L. M. Grzesiak, K. Karwowski, and A. Wawrzak. 2013. Application of ethernet powerlink for communication in a linux RTAI open CNC system. *IEEE Transactions on Industrial Electronics* 60, 2 (July 2013), 628–636.

[28] J. Eveleens. 2014. Ethernet AVB overview and status. In *Proceedings of the SMPTE 2014 Annual Technical Conference Exhibition*. 1–11.

[29] J. Falk, F. Dürr, and K. Rothermel. 2018. Exploring practical limitations of joint routing and scheduling for TSN with ILP. In *Proceedings of the 2018 IEEE 24th International Conference on Embedded and Real-Time Computing Systems and Applications*. 136–146.

[30] M. H. Farzaneh and A. Knoll. 2016. An ontology-based Plug-and-Play approach for in-vehicle Time-Sensitive Networking (TSN). In *Proceedings of the 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference*. 1–8.

[31] M. H. Farzaneh, S. Kugele, and A. Knoll. 2017. A graphical modeling tool supporting automated schedule synthesis for time-sensitive networking. In *Proceedings of the 22nd IEEE International Conference on Emerging Technologies and Factory Automation*. 1–8.

[32] M. H. Farzaneh, S. Shafaei, and A. Knoll. 2016. Formally verifiable modeling of in-vehicle time-sensitive networks (TSN) based on logic programming. In *Proceedings of the 2016 IEEE Vehicular Networking Conference*. 1–4.

[33] Christoph Fischer, Dennis Krummacker, Michael Karrenbauer, and Hans Dieter Schotten. 2020. A modular design concept for shaping future wireless TSN solutions. *Information* 12, 1 (Dec. 2020), 12.

[34] Geoffrey M. Garner, Feifei Feng, K. Den Hollander, Hongkyu Jeong, Byungsuk Kim, Byoungjoon Lee, Taechul Jung, and Jinoo Joung. 2007. IEEE 802.1 AVB and its application in carrier-grade ethernet [standards topics]. *IEEE Communications Magazine* 45, 12 (Dec. 2007), 126–134.

[35] V. Gavriluţ and P. Pop. 2018. Scheduling in time sensitive networks (TSN) for mixed-criticality industrial applications. In *Proceedings of the 2018 14th IEEE International Workshop on Factory Communication Systems*. 1–4.

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

[37] Voica Gavrilut, Bahram Zarrin, Paul Pop, and Soheil Samii. 2017. Fault-tolerant topology and routing synthesis for IEEE time-sensitive networking. In *Proceedings of the 25th International Conference on Real-Time Networks and Systems*. 267–276.

[38] E. Genc and L. F. Del Carpio. 2019. Wi-Fi QoS enhancements for downlink operations in industrial automation using TSN. In *Proceedings of the 2019 15th IEEE International Workshop on Factory Communication Systems*. 1–6.

[39] F. Geyer, E. Heidinger, S. Schneele, and A. von Bodisco. 2013. Evaluation of audio/video bridging forwarding method in an avionics switched ethernet context. In *Proceedings of the 2013 IEEE Symposium on Computers and Communications*. 000711–000716.

[40] F. Groß, T. Steinbach, F. Korf, T. C. Schmidt, and B. Schwarz. 2014. A hardware/software co-design approach for Ethernet controllers to support time-triggered traffic in the upcoming IEEE TSN standards. In *Proceedings of the 2014 IEEE 4th International Conference on Consumer Electronics Berlin*. 9–13.

[41] M. Gundall, C. Huber, P. Rost, R. Halfmann, and H. D. Schotten. 2020. Integration of 5G with TSN as prerequisite for a highly flexible future industrial automation: Time synchronization based on IEEE 802.1AS. In *Proceedings of the IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*. 3823–3830.

[42] M. Haeberle, F. Heimgaertner, H. Loehr, N. Nayak, D. Grewe, S. Schildt, and M. Menth. 2020. Softwarization of automotive E/E architectures: A software-defined networking approach. In *Proceedings of the 2020 IEEE Vehicular Networking Conference*. 1–8.

[43] Feng He, Lin Zhao, and Ershuai Li. 2017. Impact analysis of flow shaping in Ethernet-AVB/TSN and AFDX from network calculus and simulation perspective. *Sensors* 17, 5 (May 2017), 1181.

[44] E. Heidinger, F. Geyer, S. Schneele, and M. Paulitsch. 2012. A performance study of audio video bridging in aeronautic Ethernet networks. In *Proceedings of the 7th IEEE International Symposium on Industrial Embedded Systems*. 67–75.

[45] D. Hellmanns, J. Falk, A. Glavackij, R. Hummen, S. Kehrer, and F. Dürr. 2020. On the performance of stream-based, class-based time-aware shaping and frame preemption in TSN. In *Proceedings of the 2020 IEEE International Conference on Industrial Technology*. 298–303.

[46] C. Herber, A. Richter, T. Wild, and A. Herkersdorf. 2015. Real-time capable CAN to AVB ethernet gateway using frame aggregation and scheduling. In *Proceedings of the 2015 Design, Automation Test in Europe Conference Exhibition (DATE)*. 61–66.

[47] John Hershberger, Matthew Maxel, and Subhash Suri. 2007. Finding the k shortest simple paths: A new algorithm and its implementation. *ACM Transactions on Algorithms (TALG)* 3, 4 (Nov. 2007), 45–es.

[48] 2011. IEEE 802.1AS-2011 - IEEE Standard for Local and Metropolitan Area Networks - Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks. Retrieved December 10, 2020 from https://standards.ieee.org/standard/802_1AS-2011.html.

[49] IEEE. 802.1AS-Rev—Timing and Synchronization for Time-Sensitive Applications. Retrieved December 10, 2020 from https://1.ieee802.org/tsn/802-1as-rev/.

[50] 2017. IEEE 802.1CB-2017 - IEEE Standard for Local and metropolitan area networks–Frame Replication and Elimination for Reliability. Retrieved December 10, 2020 from https://standards.ieee.org/standard/802_1CB-2017.html.

[51] 2010. IEEE 802.1Qat-2010 - IEEE Standard for Local and metropolitan area networks–Virtual Bridged Local Area Networks Amendment 14: Stream Reservation Protocol (SRP). Retrieved December 10, 2020 from https://standards.ieee.org/standard/802_1Qat-2010.html.

[52] 2009. IEEE 802.1Qav-2009 - IEEE Standard for Local and metropolitan area networks– Virtual Bridged Local Area Networks Amendment 12: Forwarding and Queuing Enhancements for Time-Sensitive Streams. Retrieved December 10, 2020 from https://standards.ieee.org/standard/802_1Qav-2009.html.

[53] 2016. IEEE 802.1Qbu-2016 - IEEE Standard for Local and metropolitan area networks – Bridges and Bridged Networks – Amendment 26: Frame Preemption. Retrieved December 10, 2020 from https://standards.ieee.org/standard/802_1Qbu-2016.html.

[54] IEEE. 802.1Qbv-2015—Enhancements for Scheduled Traffic. Retrieved from https://standards.ieee.org/standard/802_1Qbv-2015.html.

[55] 2015. IEEE 802.1Qca-2015 - IEEE Standard for Local and metropolitan area networks– Bridges and Bridged Networks - Amendment 24: Path Control and Reservation. Retrieved December 10, 2020 from https://standards.ieee.org/standard/802_1Qca-2015.html.

[56] 2017. IEEE 802.1Qci-2017 - IEEE Standard for Local and metropolitan area networks–Bridges and Bridged Networks–Amendment 28: Per-Stream Filtering and Policing. Retrieved December 10, 2020 from https://standards.ieee.org/standard/802_1Qci-2017.html.

[57] IEEE. IEEE 802.1 Working Group. Retrieved December 10, 2020 from https://1.ieee802.org/.

[58] IEEE. P802.1DG—TSN Profile for Automotive In-Vehicle Ethernet Communications. Retrieved December 10, 2020 from https://1.ieee802.org/tsn/802-1dg/.

[59] IEEE. Time-Sensitive Networking (TSN) Task Group. Retrieved December 10, 2020 from https://1.ieee802.org/tsn/.

[60] J. Imtiaz, J. Jasperneite, and L. Han. 2009. A performance study of Ethernet Audio Video Bridging (AVB) for Industrial real-time communication. In *Proceedings of the 2009 IEEE Conference on Emerging Technologies Factory Automation*. 1–8.

[61] Andreas Kern, Helge Zinner, Thilo Streichert, Josef Nöbauer, and Jürgen Teich. 2011. Accuracy of Ethernet AVB time synchronization under varying temperature conditions for automotive networks. In *Proceedings of the 48th Design Automation Conference*. Association for Computing Machinery, 597–602.

[62] O. Kleineberg, P. Fröhlich, and D. Heffernan. 2012. Fault-tolerant Audio and Video Bridging (AVB) Ethernet: A novel method for redundant stream registration configuration. In *Proceedings of the 2012 IEEE 17th International Conference on Emerging Technologies Factory Automation*. 1–8.

[63] J. Ko, J. Lee, C. Park, and S. Park. 2015. Research on optimal bandwidth allocation for the scheduled traffic in ieee 802.1 AVB. In *Proceedings of the 2015 IEEE International Conference on Vehicular Electronics and Safety*. 31–35.

[64] Sune Mølgaard Laursen, Paul Pop, and Wilfried Steiner. 2016. Routing optimization of AVB streams in TSN networks. *ACM Sigbed Review* 13, 4 (Nov. 2016), 43–48.

[65] Juho Lee and Sungkwon Park. 2019. Time-Sensitive Network (TSN) experiment in sensor-based integrated environment for autonomous driving. *Sensors* 19, 5 (March 2019), 1111.

[66] Kang B. Lee and J. Eldson. 2004. Standard for a precision clock synchronization protocol for networked measurement and control systems. In *Proceedings of the 2004 Conference on IEEE 1588, Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*.

[67] Y. S. Lee, J. H. Kim, and J. W. Jeon. 2017. FlexRay and ethernet AVB synchronization for high QoS automotive gateway. *IEEE Transactions on Vehicular Technology* 66, 7 (Dec. 2017), 5737–5751.

[68] L. Leonardi, L. L. Bello, and G. Patti. 2020. Performance assessment of the IEEE 802.1Qch in an automotive scenario. In *Proceedings of the 2020 AEIT International Conference of Electrical and Electronic Technologies for Automotive (AEIT AUTOMOTIVE)*. 1–6.

[69] Xiaoting Li and Laurent George. 2017. Deterministic delay analysis of AVB switched Ethernet networks using an extended Trajectory Approach. *Real-Time Systems* 53, 1 (Jan. 2017), 121–186.

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

[71] Hyung-Taek Lim, Daniel Herrscher, Martin Johannes Waltl, and Firas Chaari. 2012. Performance analysis of the IEEE 802.1 ethernet audio/video bridging standard. In *Proceedings of the International ICST Conference on Simulation Tools and Techniques*. 27–36.

[72] C. Lin and H. Yu. 2016. Invited: Cooperation or competition? Coexistence of safety and security in next-generation Ethernet-based automotive networks. In *Proceedings of the 2016 53rd ACM/EDAC/IEEE Design Automation Conference*. 1–6.

[73] X. Liu, Z. Nie, D. Li, and H. Yu. 2019. Design of an improved ethernet AVB model for real-time communication in in-vehicle network. In *Proceedings of the 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference*. 6–10.

[74] L. Lo Bello, E. Bini, and G. Patti. 2015. Priority-driven swapping-based scheduling of aperiodic real-time messages over EtherCAT networks. *IEEE Transactions on Industrial Informatics* 11, 3 (Aug. 2015), 741–751.

[75] Lucia Lo Bello, Gaetano Patti, and Giancarlo Vasta. 2021. Assessments of real-time communications over TSN automotive networks. *Electronics* 10, 5 (Feb. 2021), 556.

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

[77] I. Álvarez, M. Barranco, and J. Proenza. 2018. Towards a fault-tolerant architecture based on time sensitive networking. In *Proceedings of the 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation*, Vol. 1. 1113–1116.

[78] I. Álvarez, J. Proenza, M. Barranco, and M. Knezic. 2017. Towards a time redundancy mechanism for critical frames in time-sensitive networking. In *Proceedings of the 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation*. 1–4.

[79] Rouhollah Mahfouzi, Amir Aminifar, Soheil Samii, Petru Eles, and Zebo Peng. 2019. Security-aware routing and scheduling for control applications on Ethernet TSN networks. *ACM Transactions on Design Automation of Electronic Systems* 25, 1 (Nov. 2019), 1–26.

[80] T. L. Mai, N. Navet, and J. Migge. 2019. A hybrid machine learning and schedulability analysis method for the verification of TSN networks. In *Proceedings of the 2019 15th IEEE International Workshop on Factory Communication Systems*. 1–8.

[81] 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.

[82] Jörn Migge, Josetxo Villanueva, Nicolas Navet, and Marc Boyer. 2018. Insights on the performance and configuration of AVB and TSN in automotive ethernet networks. In *Proceedings of the 9th European Congress on Embedded Real Time Software and Systems*.

[83] Daniel Minoli and Benedict Occhiogrosso. 2018. Blockchain mechanisms for IoT security. *Internet of Things* 1 (Sept. 2018), 1–13.

[84] S. Mohalik, D. B. Chokshi, M. G. Dixit, A. C. Rajeev, and S. Ramesh. 2013. Scalable model-checking for precise end-to-end latency computation. In *Proceedings of the 2013 IEEE Conference on Computer Aided Control System Design*. 19–24.

[85] A. Nasrallah, A. S. Thyagaturu, Z. Alharbi, C. Wang, X. Shao, M. Reisslein, and H. Elbakoury. 2019. Performance comparison of IEEE 802.1 TSN Time Aware Shaper (TAS) and Asynchronous Traffic Shaper (ATS). *IEEE Access* 7 (April 2019), 44165–44181.

[86] A. Nasrallah, A. S. Thyagaturu, Z. Alharbi, C. Wang, X. Shao, M. Reisslein, and H. ElBakoury. 2019. Ultra-low latency (ULL) networks: The IEEE TSN and IETF DetNet standards and related 5G ULL research. *IEEE Communications Surveys Tutorials* 21, 1 (Sept. 2019), 88–145.

[87] M. A. Ojewale, P. M. Yomsi, and B. Nikolić. 2020. Multi-level preemption in TSN: Feasibility and requirements analysis. In *Proceedings of the 2020 IEEE 23rd International Symposium on Real-Time Distributed Computing*. 47–55.

[88] Mubarak Adetunji Ojewale, Patrick Meumeu Yomsi, and Borislav Nikolić. 2021. Worst-case traversal time analysis of TSN with multi-level preemption. *Journal of Systems Architecture* 116 (March 2021), 102079.

[89] M. Pahlevan and R. Obermaisser. 2018. Evaluation of time-triggered traffic in time-sensitive networks using the OPNET simulation framework. In *Proceedings of the 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing*. 283–287.

[90] M. Pahlevan and R. Obermaisser. 2018. Genetic algorithm for scheduling time-triggered traffic in time-sensitive networks. In *Proceedings of the 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation*, Vol. 1. 337–344.

[91] J. Park, B. Cheoun, and J. Jeon. 2015. Worst-case analysis of ethernet AVB in automotive system. In *Proceedings of the 2015 IEEE International Conference on Information and Automation*. 1696–1699.

[92] Paul Pop, Michael Lander Raagaard, Silviu S. Craciunas, and Wilfried Steiner. 2016. Design optimisation of cyber-physical distributed systems using IEEE time-sensitive networks. *IET Cyber-Physical Systems: Theory and Applications* 1, 1 (Dec. 2016), 86–94.

[93] F. Prinz, M. Schoeffler, A. Lechler, and A. Verl. 2018. End-to-end redundancy between real-time I4.0 components based on time-sensitive networking. In *Proceedings of the 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation*, Vol. 1. 1083–1086.

[94] Shifei Qian, Feng Luo, and Jinpeng Xu. 2017. An analysis of frame replication and elimination for time-sensitive networking. In *Proceedings of the 2017 VI International Conference on Network, Communication and Computing*. 166–170.

[95] R. Queck. 2012. Analysis of ethernet AVB for automotive networks using network calculus. In *Proceedings of the 2012 IEEE International Conference on Vehicular Electronics and Safety*. 61–67.

[96] Michael Lander Raagaard and Paul Pop. 2017. *Optimization Algorithms for the Scheduling of IEEE 802.1 Time-Sensitive Networking (TSN)*. Technical Report. Technical University of Denmark. Retrieved December 10, 2020 from http://www2.compute.dtu.dk/~paupo/publications/Raagaard2017aa-Optimization%20algorithms%20for%20th-.pdf.

[97] M. L. Raagaard, P. Pop, M. Gutiérrez, and W. Steiner. 2017. Runtime reconfiguration of time-sensitive networking (TSN) schedules for Fog Computing. In *Proceedings of the 2017 IEEE Fog World Congress*. 1–6.

[98] F. Reimann, S. Graf, F. Streit, M. Glaß, and J. Teich. 2013. Timing analysis of Ethernet AVB-based automotive E/E architectures. In *Proceedings of the 2013 IEEE 18th Conference on Emerging Technologies Factory Automation*. 1–8.

[99] Dominik Reinhardt and Markus Kucera. 2013. Domain controlled architecture—a new approach for large scale software integrated automotive systems. In *Proceedings of the 3rd International Conference on Pervasive and Embedded Computing and Communication Systems*.

[100] H. J. Rivera Verduzco, P. J. L. Cuijpers, and J. Cao. 2017. Work-in-progress: Best-case response time analysis for ethernet AVB. In *Proceedings of the 2017 IEEE Real-Time Systems Symposium*. 378–380.

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

[102] S. Rumpf, T. Steinbach, F. Korf, and T. C. Schmidt. 2014. Software stacks for mixed-critical applications: Consolidating IEEE 802.1 AVB and time-triggered ethernet in next-generation automotive electronics. In *Proceedings of the 2014 IEEE 4th International Conference on Consumer Electronics Berlin*. 14–18.

[103] A. Sabry, A. Omar, M. Hammad, and N. Abdelbaki. 2020. AVB/TSN protocols in automotive networking. In *Proceedings of the 15th International Conference on Computer Engineering and Systems*. 1–7.

[104] S. Samii and H. Zinner. 2018. Level 5 by layer 2: Time-sensitive networking for autonomous vehicles. *IEEE Communications Standards Magazine* 2, 2 (July 2018), 62–68.

[105] Jürgen Scheuring and Ulrich Bernhard. 2020. Time-sensitive networking in the automotive industry. *ATZelectronics Worldwide* 15, 9 (Sept. 2020), 54–58.

[106] S. Schneele and F. Geyer. 2012. Comparison of IEEE AVB and AFDX. In *Proceedings of the 2012 IEEE/AIAA 31st Digital Avionics Systems Conference*. 1–9.

[107] Luca Sciullo, Sushmit Bhattacharjee, and Matthias Kovatsch. 2020. Bringing deterministic industrial networking to the W3C web of things with TSN and OPC UA. In *Proceedings of the 10th International Conference on the Internet of Things*. Association for Computing Machinery, Article 9, 8 pages.

[108] Maik Seewald and Robert Barton. 2019. Integrated security-tagging for deterministic ethernet. (Feb. 2019). Retrieved December 10, 2020 from https://www.tdcommons.org/dpubs_series/1969.

[109] G. S. Sestito, A. C. Turcato, A. L. Dias, M. S. Rocha, M. M. da Silva, P. Ferrari, and D. Brandao. 2018. A method for anomalies detection in real-time ethernet data traffic applied to PROFINET. *IEEE Transactions on Industrial Informatics* 14, 5 (Nov. 2018), 2171–2180.

[110] D. Shrestha, Z. Pang, and D. Dzung. 2018. Precise clock synchronization in high performance wireless communication for time sensitive networking. *IEEE Access* 6 (Feb. 2018), 8944–8953.

[111] L. Silva, P. Pedreiras, P. Fonseca, and L. Almeida. 2019. On the adequacy of SDN and TSN for Industry 4.0. In *Proceedings of the 2019 IEEE 22nd International Symposium on Real-Time Distributed Computing*. 43–51.

[112] F. Smirnov, M. Glaß, F. Reimann, and J. Teich. 2017. Formal timing analysis of non-scheduled traffic in automotive scheduled TSN networks. In *Proceedings of the Design, Automation Test in Europe Conference Exhibition*. 1643–1646.

[113] T. Steinbach, H. Lim, F. Korf, T. C. Schmidt, D. Herrscher, and A. Wolisz. 2012. Tomorrow's in-car interconnect? A competitive evaluation of IEEE 802.1 AVB and time-triggered ethernet (AS6802). In *Proceedings of the 2012 IEEE Vehicular Technology Conference*. 1–5.

[114] W. Steiner, P. Heise, and S. Schneele. 2014. Recent IEEE 802 developments and their relevance for the avionics industry. In *Proceedings of the 2014 IEEE/AIAA 33rd Digital Avionics Systems Conference*. 1–12.

[115] S. Thangamuthu, N. Concer, P. J. L. Cuijpers, and J. J. Lukkien. 2015. Analysis of ethernet-switch traffic shapers for in-vehicle networking applications. In *Proceedings of the 2015 Design, Automation Test in Europe Conference Exhibition*. 55–60.

[116] D. Thiele and R. Ernst. 2016. Formal worst-case timing analysis of Ethernet TSN's burst-limiting shaper. In *Proceedings of the 2016 Design, Automation Test in Europe Conference Exhibition*. 187–192.

[117] D. Thiele, R. Ernst, and J. Diemer. 2015. Formal worst-case timing analysis of Ethernet TSN's time-aware and peristaltic shapers. In *Proceedings of the 2015 IEEE Vehicular Networking Conference*. 251–258.

[118] K. S. Umadevi and R. K. Sridharan. 2017. Multilevel ingress scheduling policy for time sensitive networks. In *Proceedings of the 2017 International Conference on Microelectronic Devices, Circuits and Systems*. 1–4.

[119] Yang Wang, Jidong Chen, Wei Ning, Hao Yu, Shimei Lin, Zhidong Wang, Guanshi Pang, and Chao Chen. 2020. A time-sensitive network scheduling algorithm based on improved ant colony optimization. *Alexandria Engineering Journal* 60, 1 (June 2020), 107–114.

[120] M. Wollschlaeger, T. Sauter, and J. 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 (March 2017), 17–27.

[121] Jinpeng Xu, Feng Luo, and Hailun Shao. 2016/11. Failure rate analysis for time-sensitive networking. In *Proceedings of the 2016 5th International Conference on Measurement, Instrumentation and Automation*. Atlantis Press.

[122] Q. Yu and M. Gu. 2020. Adaptive group routing and scheduling in multicast time-sensitive networks. *IEEE Access* 8 (Feb. 2020), 37855–37865.

[123] Q. Yu, H. Wan, X. Zhao, Y. Gao, and M. Gu. 2020. Online scheduling for dynamic VM migration in multicast time-sensitive networks. *IEEE Transactions on Industrial Informatics* 16, 6 (June 2020), 3778–3788.

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

[125] Luxi Zhao, Paul Pop, Zhong Zheng, Hugo Daigmorte, and Marc Boyer. 2018. Improving worst-case end-to-end delay analysis of multiple classes of AVB traffic in TSN networks using network calculus. Retrieved December 10, 2020 from https://www.semanticscholar.org/paper/.

[126] Luxi Zhao, Paul Pop, Zhong Zheng, Hugo Daigmorte, and Marc Boyer. 2021. Latency analysis of multiple classes of AVB traffic in TSN with standard credit behavior using network calculus. *IEEE Transactions on Industrial Electronics* 68, 10 (Oct. 2021), 10291–10302. https://doi.org/10.1109/TIE.2020.3021638

[127] L. Zhao, P. Pop, Z. Zheng, and Q. Li. 2018. Timing analysis of AVB traffic in TSN networks using network calculus. In *Proceedings of the 2018 IEEE Real-Time and Embedded Technology and Applications Symposium*. 25–36.

[128] Mingxiong Zhao, Jun-Jie Yu, Wen-Tao Li, Di Liu, Shaowen Yao, Wei Feng, Changyang She, and Tony QS Quek. 2020. Energy-aware offloading in time-sensitive networks with mobile edge computing. arXiv:2003.12719. Retrieved from https://arxiv.org/abs/2003.12719.

Received December 2020; revised September 2021; accepted September 2021