Key Technologies of Time-Sensitive Networking and Its Application

Qing Li1, 2, 3, 4, a, Dong Li1, 2, 3, * and Xinbo Sun1, 2, 3, 4, b

1Key Laboratory of Networked Control System, Chinese Academy of Sciences, Shenyang 110016, China
2Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China
3Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang 110016, China
4University of Chinese Academy of Sciences, Beijing 100049, China

*Corresponding author e-mail: lidong@sia.cn, a liqing@sia.cn, b15524011088@163.com

Abstract. Network is the core element of many industrial and automation systems. In the future, more attention will be paid to convergent and unified network. Time-critical traffic and non-time-critical traffic need to share communication channels. The goal of the IEEE TSN task group is to extend the existing Ethernet standard to achieve a degree of certainty that meets the hard-real-time requirements of modern control networks in the industrial automation and automotive industries. Time sensitive network (TSN) is characterized by low jitter, low delay and deterministic transmission. It can transmit critical and non-critical traffic in the same network, which is very suitable for time-critical applications with high requirements on transmission delay. This paper briefly summarizes the key components of TSN, studies the key technologies, and carries out a simple traffic scheduling experiment on the switch based on 802.1Qbv function, and finally analyzes the application of TSN in industrial automation and automobile industry.

1. Introduction

In recent years, with the development of Time-Sensitive market, such as automotive and industrial automation, the demand for deterministic real-time communication is becoming stronger. It is necessary to ensure that specific data is received within a strict time interval. Therefore, distributed safety-critical applications in these areas need bounded low latency and low jitter. Standard Ethernet IEEE802.3 [1] providing bandwidth requirement and physical layer options for a wide range of application field, while maintaining scalability and cost-effectiveness, has been widely used in LAN and WANs. But it is not suitable for real-time and safety-critical applications, therefore, several real-time Ethernet variants, such as specification for ARINC 664-part7 (AFDX), TT-Ethernet, ProfiNET, etc., have been proposed, and recommended and used in industry. However, they are incompatible with each other, therefore, they cannot run on the same physical link in the network without losing real-time guarantee [2]. Time is not the only key aspect to consider, the reliability, fault tolerance, and security are other key demand for these applications. IEEE Time-Sensitive Networking (TSN) task group [3] is working on these problems.
IEEE TSN task group was founded in 2012 which renamed from Audio Video Bridging (AVB) task group [4]. TSN is a set of IEEE 802 Ethernet sub-standards defined by IEEE TSN task group which support deterministic real-time communication through standard Ethernet. It provides services of bridging, network management and building real-time transmission over Ethernet within a LAN or MAN domain. TSN has a high degree of time-sensitive, which can guarantee the worst-case end-to-end delay, high reliability, bandwidth isolation and zero congestion loss [5].

Time-sensitive communication occurs through so-called time-sensitive flows, defined by the sender (or speaker) and one or more receivers (or listeners). Traffic flows in a network often outnumber traffic classes, so streams must share traffic classes. TSN offers several traffic types, such as time-triggered (TT) traffic, audio-video bridging (AVB) traffic, and best-effort (BE) traffic, which are ideal for applications with different security critical levels (high-critical, mission-critical, and non-critical). TT traffic supports hard real-time applications that require very low latency and jitter. It has the highest priority, and are sent based on a schedule (called Gate Control List (GCL)) proposed in 802.1Qbv [6], which relies on 802.1 AS-Rev global synchronization clock [7]. AVB is suitable for applications that requiring bounded end-to-end delay. It is an asynchronous communication type whose priority is lower than TT communication. It uses Credit Based Shaper (CBS) as defined in IEEE 802.1BA [8] to prevent the starvation of low priority flow. BE traffic has the lowest priority and is suitable for applications that do not require any timing guarantees. The choice of message traffic classes depends on the specificity of the application.

2. TSN overview

2.1. Benefits of TSN

TSN is an open and standard technology and does not belong to any organization or company. For a long time, industrial automation market has been faced with the problem of incompatibility of various fieldbuses. TSN has advantages in this respect, as follows:

- TSN guarantees the compatibility of devices from different suppliers at the network level. The vendors of each controller usually promotes a specific fieldbus system. For users, this means that the choice of controller basically determines the choice of bus. Therefore, because different bus systems are not compatible with each other, user often end up relying on the manufacturer.

- TSN is a part of Ethernet standard series, so it can be naturally extended to Ethernet, which makes TSN not limited in bandwidth or other Ethernet performance standards. New nodes can be easily added to the network and discovered through standard protocols.

- TSN can break down communication barriers between critical systems and non-critical systems, allowing the convergence of network and system. Critical traffic and non-critical traffic can share communication channel, which can ensure the timing of critical traffic and temporal isolation of non-critical communication.

- Higher level protocols can be combined with TSN, because TSN is completely implemented in the data link layer (OSI model 2). The convergence of TSN and Open Platform Communication Unified Architecture (OPC UA) is a hot topic nowadays. OPC UA is used to solve the interoperability problem, while TSN is used to solve the physical unity problem of communication.

2.2. Bridges and Bridged Networks

IEEE 802.1Q Bridges and Bridged Networks [9] is a network standard supporting virtual LAN (VLAN) on IEEE 802.3 Ethernet. The standard defines VLAN tagging system for Ethernet frames, and the related processes of bridges and switches when processing these frames.

1) Priority scheme: Scheduling and traffic shaping allow different traffic classes with different priorities to coexist on the same network, and each traffic class has different requirements for end-to-end latency and available bandwidth. According to IEEE 802.1Q standard bridging uses eight different priority and strict priority scheme (as shown in Fig. 1).
2) VLAN tag: Compared with traditional Ethernet, TSN-based networks allow four bytes to be added to the Ethernet data frame header to define its characteristics, as shown in Fig. 2.

- Tag Protocol Identifier: Network type identification, representing that this is a TSN network, marked 0x8100.
- Priority Code Point: A 3-bit field, ranging from 0 to 7, are used as the priority of the data flow.
- Drop Eligible Indicator: A 1-bit field, 0 for standard format, 1 for non-standard format.
- VLAN Identifier: A 12-bit field, assigning the frame to a particular VLAN. A value of 0 indicates that the frame does not belong to any VLAN. 0x000 and 0xFFF are reserved values, other values can be used as VLAN identifiers, 4094 in all. It indicates that TSN is designed for large-scale data transmission.

| Background | Best Effort | Excellent Effort | Critical Applications | Video <100ms latency | Audio <10ms latency | Internetwork Control | Network Control |
|------------|-------------|------------------|----------------------|----------------------|----------------------|---------------------|------------------|
| 1          | 0           | 2                | 3                    | 4                    | 5                    | 6                   | 7                |

**Figure 1.** Traffic class in 802.1Q.

2.3. **TSN Components**

TSN standard set can be thought of as a toolbox containing several valuable tools, which can be categorized into four groups: time synchronization, bounded low latency, resource management, and reliability (see Fig.3, The standard with the letter P is under development). Each standard specification in TSN can be used independently, and is basically self-sufficient. However, they are not unrelated, and some standards can be used in many aspects. For instance, IEEE 802.1AS-Rev solves the time synchronization problem, but it is related to reliability and latency.

2.3.1. **Time Synchronization.** Time plays a crucial role in the TSN network. For real-time communication with strict time requirements, all devices in the network need to have a common time reference and synchronize their clocks with each other. Only through synchronized clocks, it is possible for all network devices run uniformly, and can accurately perform the required operations at the required time point, so that end stations and bridges can synchronize their local clocks with each other. In TSN network, time is usually allocated from a central time source directly through the network itself. In most cases, this is done by using the IEEE1588 Precision Time Protocol [10].

IEEE 1588 standard defines the Precision Time Protocol (PTP) for synchronizing clocks across an entire network. IEEE802.1AS standard is a profile of IEEE 1588 specified by the IEEE TSN task group. It provides accurate synchronization of network node time and achieves a reference time accuracy better
than 1 μs. IEEE802.1AS-Rev is creating a profile of the IEEE 1588 PTP synchronization protocol for TSN, will enable clock synchronization compatibility between different TSN devices, and has also resolved support for fault tolerance and multiple active synchronization masters. The goal of IEEE802.1AS-Rev is to maintain synchronization times for time-sensitive applications during normal network operation and after adding, deleting or network component failure and refactoring.

2.3.2. Bounded Low Latency. One of the major improvements of AVB and TSN to IEEE 802.1 networks is to ensure the delivery of messages with real-time constraints. TSN guarantees the worst-case latency of critical data through multiple queues and shaping technologies of switch ports and resources reserved for critical traffic. TSN also improves the real-time performance through two published standards: IEEE 802.1Qbv and IEEE 802.1Qbu [11]. 802.1Qbv provides time-based traffic shaping, which scheduling mechanism defined in the standard can control the flow periodically, providing a fixed schedule for different data traffic classes to determine the start and end time of the flow in advance. Such time-based transmission selection can only work at the network level when the appropriate synchronization mechanism (IEEE 802.1AS-Rev) is in place. IEEE 802.1Qbu standardizes a preemption mechanism that allows time-critical messages to interrupt ongoing non-time-critical transmissions. The relevant technologies in these two standards are described in sections 3.2 and 3.3.

2.3.3. Reliability. The reliability of data is very important for many application scenarios in TSN, especially in the case of errors. IEEE 802.1CB [12] and IEEE P802.1Qca [13] standards propose a mechanism that allows replication and redundant transmission of data over multiple disjunctive paths. This mechanism improves the reliability by transmitting multiple copies of the same packets on disjoint paths in the network. More significantly, the redundant attributes implemented by these mechanisms are transparent to communicating applications. IEEE 802.1Qci Per-Stream Filtering and Policing standards improve reliability by preventing bandwidth conflicts, malfunctioning and malicious behavior.

2.3.4. Resource Management. Resource reservation and management is the key to realization of deterministic network. There are several standards in TSN that contribute to bandwidth reservation and management in different ways. Resource management basis is defined by TSN configuration models (IEEE P802.1Qcc) [14]. In order to support proper path identification, registration and management, TSN proposes a set of mechanisms and interfaces in 802.1Qcc. Solutions to centrally control and manage the TSN, to schedule the time sensitive traffic using solutions similar to software defined

Figure 3. IEEE TSN Tools

| Time Synchronization | Ultra Reliability |
|----------------------|-------------------|
| Timing and Synchronization (802.1AS) includes a profile of IEEE 1588 | Frame Replication and Elimination (802.1CB) |
|                       | Path Control and Reservation (802.1Qca) |
|                       | Per-Stream Filtering and Policing (802.1Qci) |
|                       | Reliability for time sync (P802.1AS-Rev) |

| Bounded Low Latency | Resource Management |
|---------------------|---------------------|
| Credit Based Shaper (802.1Qav) | Stream Reservation Protocol (802.1Qat) |
| Frame preemption (802.3br & 802.1Qbu) | TSN configuration (802.1Qcc) |
| Scheduled Traffic (802.1Qbw) | Basic YANG (802.1Qxp) |
| Cyclic Queuing and Forwarding (802.1Qch) | Link-local Registration Protocol (P802.1CS) |
| Asynchronous Traffic Shaping (P802.1Qcr) | Resource Allocation Protocol (P802.1Qdd) |
| QoS Provisions (P802.1QD) | YANG for CFM (P802.1Qcx) |
|                       | YANG for LLDP (P802.1ABcu) |
|                       | YANG for Qbv, Qbu, and Qci (P802.1Qcw) |
|                       | YANG & MIB for FRER (P802.1Cbcv) |
|                       | Extended Stream Identification (P802.1Cbd) |
networking. Centralized Network Configuration (CNC) can be applied to the network devices (bridges), while Centralized User Configuration (CUC) can be applied to user devices (end nodes). The fully centralized configuration model follows a software-defined networking (SDN) approach; in other words, CNC and CUC provide the control plane instead of distributed protocols. In contrast, distributed control protocols are applied in the fully distributed model without CNC or CUC [15].

3. Key technologies

3.1. 802.1AS

IEEE 802.1AS defines the method of frequency synchronization and phase synchronization. The clock synchronization method between master and slave clocks is shown in Fig. 4. Suppose the phase difference between the master and slave clocks as $\Delta \theta$, the slave clock receives the master clock synchronization message Sync at $t_1$, and analysis the Sync sending time $t_0$ in the subsequent Follow_Up message, and the time difference between $t_1$ and $t_0$ is $d_{ms}$. The slave clock transmits Delay_Req message to the master clock at time $t_2$, and the master clock receives the message at time $t_3$. The time difference between $t_3$ and $t_2$ is $d_{sm}$, generally, $d_{ms}=d_{sm}$. The following two equations are true:

$$t_1 = t_0 + \Delta \theta + d_{ms}$$

$$t_3 = t_2 - \Delta \theta + d_{sm}$$

From (1) and (2):

$$\Delta \theta = \frac{(t_1-t_0)-(t_3-t_2)}{2}$$

$$d_{sm} = d_{ms} = \frac{(t_1-t_0)+(t_3-t_2)}{2}$$

Figure 4. 802.1AS Clock synchronization principle

IEEE 802.1AS-Re regulates the use of multiple grandmaster clocks and the possibility of establishing multiple connections with these grandmaster clocks. Replication of grandmaster clocks results in shorter fail-over times in cases when a grandmaster becomes faulty.

In a recent simulation study, Gutiérrez et al. [16] evaluated the impact of detailed physical parameter list on the synchronization quality of AS, and showed that the implementation details such as the PHY jitter and the clock granularity have a great impact on the time synchronization precision.
3.2. Time-Aware Shaper
In the IEEE 802.1Qbv standard, up to eight queues are defined per port for forwarding traffic, with each frame assigned to a queue based on QoS priority. To control the forwarding of queued traffic to the TSN switch port, 802.1Qbv defines a Time-Aware Shaper (TAS). TAS is essentially a Gate mechanism that dynamically enables or disables the selection of frames from egress queues based on a predefined loop schedule called Gate Control List (GCL). A timing gate can be either open or closed at a given time. When the gate opens, the frames in each queue are forwarded in FIFO order to a priority selection mechanism that determines which frames are forwarded to the physical link of the connection. Therefore, if two gates are open at the same time, force the transmission of high-priority frames according to their traffic class priority, delaying other low-priority traffic. Fig.5 gives an illustration of TAS for an output port of a node.

The control list of gate control scheduling needs to be scheduled according to the reserved information of resources. Gate control scheduling algorithm generally divides a basic cycle into many time slots, and different traffic classes are transmitted in different time slots. While the Qbv mechanism in itself is rather simple, the creation of the gate control list is rather complicated, that is, finding the right time to open and close the door. In fact, generally speaking, the synthesis of communication scheduling in real-word scenes is usually NP-complete. However, schedule synthesis is an active research field, and many research results can be used to Qbv. We can distinguish between two main types of research method synthesis communication scheduling: the first type aims to build a specialized search algorithm, for example, by deploying heuristic metaheuristic, or genetic algorithm and the second type utilize general purpose tools, such as integer linear programming (independent) or satisfiability modulo theories (SMT) solvers.

![Figure 5. A TAS for an output port.](image)

Craciunas et al. [17] studied fully deterministic scheduling problem for IEEE802.1Qbv-computation multi-hop switched networks, the key functional parameters that affect the deterministic behavior of real-time communication in 802.1Qbv environment are identified and analyzed. According to the generalized configuration of these parameters, the constraints required for off-line scheduling are derived to ensure the low and bounded jitter and deterministic end-to-end delay of critical traffic. A method of static scheduling of 802.1Qbv network equipment is proposed by using Satisfiability Modulo Theories (SMT) or Optimization Modulo Theories (OMT) solvers. Oliver et al. [18] discusses how to formalize the synthesis of communication scheduling for GCL defined in IEEE 802.1Qbv into a constrained system represented by first-order theory of array; The first-order array theory is discussed as an appropriate method to specify the constraint set of gate operations (open and close), and the size of GCL is used as the input constraint to the scheduler, so as to illustrate how to solve the satisfiability
problem of the system with a SMT solver. Dürr et al. [19] modeled IEEE 802.1Qbv time-sensitive network traffic scheduling problem as a No-wait Packet Scheduling Problem (NW-PSP), and mapped it to the well-known No-Wait Job-shop Scheduling Problem (NW-JSP) in the field of operations research. A heuristic optimization algorithm based on Tabu search and a scheduling compression technique to reduce the number of guard bands in scheduling are proposed to achieve efficient scheduling calculation.

3.3. Frame Preemption

However, frame transmission at the link layer in Ethernet has so far been non-preemption, regardless of the mediation mechanism. In a non-preemption transmission, the frame being transmitted is guaranteed to complete without interruption. Therefore, if a high-priority frame arrives while transmitting a low-priority frame, the time-critical high-priority frame is delayed by the low-priority frame. For example, in a worst-case scenario, for low-priority frames, the high-priority frames of each switch can be delayed by about 120 us on a 100 Mbit/s link. This may be too much for time-constrained control applications. The IEEE 802.3br standard solves this problem by introducing frame preemption into Ethernet. The TSN task group defines the preemption interface and module level in the IEEE 802.1Qbu standard. In general, the stream with higher priority is more sensitive to the size of delay. In frame preemption, frames are divided into high-speed frames and low-speed frames according to different time delay requirements, among which low-speed frames are also called preemption frames. The frame preemption example is shown in Fig. 6.

There are two frames in the example: HP and LP, where HP has a higher priority than LP. Frame HP arrives while LP is being transmitted. In the case of non-preemption, HP has to wait for LP to complete its transmission before it can transmit, which results in a long transmission delay for HP. In the case of frame preemption, the transmission of frame LP can be preempted so that the high-priority frame HP can be sent in advance. Due to the overhead of preemption, the first fragment of frame LP must end with CRC, followed by interframe gaps, so HP cannot be sent immediately.

IEEE 802.3br provides two MAC service interfaces: Preemptable media access control (pMAC) and Express media access control (eMAC). Preemptable frames should be transmitted through the pMAC service, while express frames through the eMAC service.

In the case of frame preemption, time-critical frames do not have to wait until the end of all non-critical data transmission, thus reducing the delay caused by interference of non-critical frames. However, the advantage of preemption is related to the transmission speed, and the lower the transmission speed, the greater the advantage of preemption. Because the delay caused by non-time-critical frames will be shortened at high speed, and preemption will incur some overhead, the advantages gained from preemption will be reduced. Because frame preemption mechanism changes the data frame transmission process of traditional Ethernet, it can only take effect when the devices at both ends of the network link support the mechanism.
Research on frame preemption. Lee et al. [20] proposed a frame preemption method that does not interrupt the non-critical frame being transmitted, but just preempt the frame being queued for transmission, and shorten the frame preemption method of guard band as far as possible. This study reduced the guard band to 20 bytes, effectively reducing the average delay of traffic transmission. However, there are some problems with this method. First, it has a large overhead, an identifier is added in the frame header during preemption. The second is buffer latency, because a buffer was added to process preemption. Based on 802.1Qbu-2016 and 802.3br-2016 standards, Park et al. [21] studies the comprehensive problem of frame preemption. The first is the priority assignment problem for each real-time data stream, thus allocating queues. The second is whether, for each queue, the allocation flow is transmitted on the corresponding egress port on the preemtable MAC (pMAC) interface or on the express MAC (eMAC) interface. A genetic algorithm-based optimization framework is proposed based on the worst-case preemption-aware timing analysis.

3.4. Credit-Based Shaper

The credit-based shaper defines credits in bits for two separate queues, dedicated to Class A and Class B traffic. CBS is suitable for AVB stream, which can guarantee the sending of high-priority messages as soon as possible and realize the asynchronous forwarding strategy. Class A allows a maximum delay of 2ms over 7 hops. For class B the maximum delay of 50ms is guaranteed. The CBS shapes flow frames meet the requirement of one class measurement interval (Class A: 125 μs, class B: 250 μs) according to a pre-reserved bandwidth.

The CBS algorithm defines a credit value that follows certain rules for forwarding incoming Ethernet frames. The CBS function is shown in Fig.7, starting with a credit value of 0.

1) When the egress port is idle, there is no message transmission in the channel, and credit is zero or positive, a frame will be transmitted. during transmission the credit decreases at a rate called sendSlope. See interval $[t_2, t_3]$.

$$\text{sendSlope} = \text{idleslope} - \text{PortTransmitRate} \quad (5)$$

2) When no message is transport in the channel, and negative for credit, credit increase at a rate idleSlope. See interval $[t_3, t_4]$.

$$\text{idleSlope} = \frac{\text{Reserved Bytes}}{\text{Measured Interval}} \quad (6)$$

3) The credit increases at a rate called idleSlope if frames are waiting for other queues to be transmitted. In this case, the value of credit is no longer bounded by 0. See interval $[t_4, t_2]$.

4) If the CBS buffer is empty, the credit value will remain at or be reduced to 0.
4. Experiment and application

4.1. Scheduling Experiment

We built an experimental environment with two TSN switches (port bandwidth 1000Mbps), and the network topology of the experimental environment is shown in Fig. 8. The network has five end systems $ES_0, ES_1, ES_2, ES_3,$ and $ES_4$ interconnected via network switches $SW_1$ and $SW_2$, there are two flows $flow_1$ and $flow_2$. Where $ES_0$ is the control terminal, which is used to modify the configuration file of the switch.

The experiment is to control the time: schedule two flows on different time slots. Because both these flows share the same link between $SW_1$ and $SW_2$, they are bottlenecked and competing for the 1000Mbps total bandwidth of that link. When run alone without any restriction, both can achieve 954Mbps bandwidth.

In the experiment, the priority of $flow_1$ is set to 7 (high priority), and the priority of $flow_2$ is set to 0 (low priority). The user defines how many clock ticks each time slot (called subschedule) takes, and also which flows are allowed to dequeue packets on each time slot. An internal clock generates ticks each 200ns, set the cycle to 2ms (200ns*10000). According to section 3.2, each switch port has eight priority queues, and the GCLs can be configured to control the gate opening (o)/closing (c) of each

---

4. Experiment and application

4.1. Scheduling Experiment

We built an experimental environment with two TSN switches (port bandwidth 1000Mbps), and the network topology of the experimental environment is shown in Fig. 8. The network has five end systems $ES_0, ES_1, ES_2, ES_3,$ and $ES_4$ interconnected via network switches $SW_1$ and $SW_2$, there are two flows $flow_1$ and $flow_2$. Where $ES_0$ is the control terminal, which is used to modify the configuration file of the switch.

The experiment is to control the time: schedule two flows on different time slots. Because both these flows share the same link between $SW_1$ and $SW_2$, they are bottlenecked and competing for the 1000Mbps total bandwidth of that link. When run alone without any restriction, both can achieve 954Mbps bandwidth.

In the experiment, the priority of $flow_1$ is set to 7 (high priority), and the priority of $flow_2$ is set to 0 (low priority). The user defines how many clock ticks each time slot (called subschedule) takes, and also which flows are allowed to dequeue packets on each time slot. An internal clock generates ticks each 200ns, set the cycle to 2ms (200ns*10000). According to section 3.2, each switch port has eight priority queues, and the GCLs can be configured to control the gate opening (o)/closing (c) of each
queue. When multiple gates are opened in the same slot, the forwarding is carried out according to priority. We conducted the following three experiments respectively:

1) Experiment 1. Each cycle is divided into two time slots T1 and T2. The first time slot T1 is set to 0.4ms (200ns*2000), and the second time slot T2 is set to 1.6ms (200ns*8000). Configuration of GCLs: T1: oCCCCCCC; T2: CCCCCCo.

2) Experiment 2. The configuration of GCLs is changed to: T1: oCCCCCCC; T2: oCCCCCCo.

3) Experiment 3. The configuration of GCLs is changed to: T1: oCCCCCCo; T2: CCCCCCCo.

The bandwidths of the two flows in each experiment were recorded respectively, and the experimental phenomena were shown in Table 1.

| Case       | $\text{flow}_1$ (Mbps) | $\text{flow}_2$ (Mbps) |
|------------|------------------------|------------------------|
| Experiment 1 | 191                    | 764                    |
| Experiment 2 | 953                    | 3.46                   |
| Experiment 3 | 191                    | 764                    |

The experimental results are as follows:

1) Whenever a Gate is open, packets from that Queue can be sent out the wire. In experiment 1, only Gate7 was opened in time slot T1 and only Gate0 in time slot T2. The bandwidth ratio is the same as the ratio between time slot durations. That is, $191 / (191+764) = 0.4 / (0.4+1.6) = 0.2$.

2) Experiment 2. Gate7 in time slot T1 and T2 is in the open (o) state, Gate0 in time slot T2 is in the open state, but $\text{flow}_2$ has a lower priority. Although Gate0 is opened in time slot T2, according to the priority rules, only when $\text{flow}_2$ transmission ends, $\text{flow}_2$ can be transmitted. The bandwidth ratio is the same as the ratio between time slot durations. That is, $953 / (953+3.46) \approx (0.4+1.6) / (0.4+1.6) = 1$.

3) Experiment 3. Although Gate0 is also open in time slot T1, $\text{flow}_1$ has a higher priority, so only $\text{flow}_1$ is transmitted in T1 and $\text{flow}_2$ is transmitted only in time slot T2. The bandwidth ratio is the same as the ratio between time slot durations. That is, $191 / (191+764) = 0.4 / (0.4+1.6) = 0.2$.

4.2. Industrial Automation

With the increasing interdependence of industrial systems, TSN plays a more significant role in achieving precise collaboration between automation components, and provides a factory solution with interoperability and scalability. At present, information technology (IT) and operation technology (OT) are separated in the industrial network. It is an inevitable trend for the development of industrial network to build a fully interconnected, flat and flexible industrial network architecture that integrates IT and OT. Many high-level industrial communication protocols can be combined with TSN, and OPC UA is the most reasonable choice. The combination of TSN and OPC UA can realize the integration of IT and OT [22]. Similar to TSN, OPC UA is an open and standard technology, vendor independent and very useful for industrial applications. The combination of OPC UA and TSN provides a completely open, standard and interoperable solution, which meets most industrial communication requirements.

Industrial robot is a programmable mechanical device designed to perform dangerous or repetitive tasks on behalf of a human with high accuracy. One of the key challenges in robotics is the lack of standard communication protocols. Robot manufacturers must support many specific protocols, which can lead to increased integration time and cost. Modern robots integrate artificial intelligence (AI), machine vision and predictive maintenance into a system. Therefore, sensors and actuators need to transmit high-bandwidth data in real time. A common solution is to use a specific channel for real-time control (such as EtherCAT and PROFINET) and another channel for higher-bandwidth communication (TCP/UDP). For applications that generate high bandwidth traffic (100 MB to 1 GB), this approach of using two separate communication channels has low efficiency. TSN provides a shared communication channel for high bandwidth traffic and real-time control traffic.
Run-time configuration is a problem faced by industrial automation, the article [23] advocated using TSN for fog computing in industrial automation, which proposes a configuration agent architecture based on 802.1Qcc and OPC UA, and proposes a scheduling algorithm based on heuristic, used for reconfiguration of timeliness traffic scheduling at runtime.

4.3. Automotive
With the development of electronic technology, automotive electronic system becomes more and more complex. With the development of intelligent driving technology, more and more signals need to be processed in vehicle controller. In addition, with the emergence of Advanced Driver Assistance Systems (ADAS), ADAS to improve safety-critical automotive applications is the next wave of trend, which requires a large amount of data transmission and subsequent processing, and the demand for bandwidth and cross-domain communication is also increasing. The load rate of car body communication network dominated by CAN and LIN networks is getting higher and higher, and it is difficult to deal with a large number of data of the body. Ethernet, as a kind of high-throughput network, has positive significance in the application of automobile. Due to the high real-time requirements of vehicle body system on data, traditional Ethernet is difficult to meet the real-time requirements. Therefore, Ethernet using TSN becomes an ideal network choice. TSN meets the sensor bus requirements well. The deployment of Qbv, CB, and Qci can meet the specific requirements of time-critical, high-availability, and security-critical communications. In addition, flexibility can be achieved through a centralized configuration model defined by Qcc. The recent survey in [24] provides an overview of the next generation of automotive systems and focuses on the relevant technologies applied by TSN in vehicle network.

5. Conclusion
Time sensitive network (TSN) is the umbrella term for a set of standards, which meet the requirements for hard real-time data transmission. Time sensitive network technology is still in the early stage of development. Due to the different timelines of various standards, TSN can only realize some functions now, and all the functions can only be provided in the next few years. In this paper, we introduced the basic components of TSN, studied the key technologies in the network, built an experimental verification environment with TSN switch, carried out a simple TSN traffic scheduling experiment, and finally briefly discussed the application fields. In the future work, we will pay close attention to the research and development of TSN international standards, strengthen exchanges with various enterprises, and jointly discuss the application and development of TSN.

Acknowledgments
The work was supported by the Intelligent Manufacturing Integrated Standardization and New Model Application Project, Standard Research and Test Verification of Industrial Internet Application Protocol and Data Recognition; and supported by the Project of CPS nanosecond synchronization technology, Autonomous project of the institute of robotics and intelligent manufacturing innovation, Chinese Academy of Sciences.

References
[1] 802.3-2015 - IEEE Standard for Ethernet. [Online]. Available: https://ieeexplore.ieee.org/document/7428776
[2] M. Ashjaei et al., “Improved Message Forwarding for Multi-Hop HaRTES Real-Time Ethernet Networks,” J. Signal Process. Syst., vol. 84, no. 1, pp. 47–67, 2016.
[3] Institute of Electrical and Electronics Engineers, Inc, “Time Sensitive Networking Task Group,” [Online]. Available: http://www.ieee802.org/1/pages/tns.html
[4] Institute of Electrical and Electronics Engineers, Inc,” Audio Video Bridging Task Group,” [Online]. Available: http://www.ieee802.org/1/pages/avbridges.html
[5] J. L. Messenger, “Time-sensitive networking: An introduction,” IEEE Commun. Standards Mag., vol. 2, no. 2, pp. 29–33, Jun. 2018.
[6] Institute of Electrical and Electronics Engineers, Inc, “802.1Qbv: Enhancements for Scheduled Traffic,” [Online]. Available: https://standards.ieee.org/standard/802_1Qbv-2015.html

[7] Institute of Electrical and Electronics Engineers, Inc, "802.1AS-Rev: Timing and Synchronization for Time-Sensitive Applications," [Online]. Available: http://www.ieee802.org/1/pages/802.1AS-rev.html

[8] Institute of Electrical and Electronics Engineers, Inc, 802.1BA: "Audio Video Bridging (AVB) Systems," [Online]. Available: https://standards.ieee.org/standard/802_1BA-2011.html

[9] IEEE Standard for Local and Metropolitan Area Network–Bridges and Bridged Networks, IEEE Standard 802.1Q-2018 (Revision of IEEE Standard 802.1Q-2014), Jul. 2018, pp. 1–1993. [Online]. Available: https://standards.ieee.org/standard/802_1Q-2018.html

[10] IEEE Micro, “1588-2002 IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems”, IEEE Standard

[11] Institute of Electrical and Electronics Engineers, Inc,”802.1Qbu: Frame Preemption,” [Online]. Available: https://standards.ieee.org/standard/802_1Qbu-2016.html

[12] Institute of Electrical and Electronics Engineers, Inc,”802.1CB: Frame Replication and Elimination for Reliability,” [Online]. Available: https://standards.ieee.org/standard/802_1CB-2017.html

[13] Institute of Electrical and Electronics Engineers, Inc,”802.1Qca: Path Control and Reservation,” [Online]. Available: https://standards.ieee.org/standard/802_1Qca-2015.html

[14] Institute of Electrical and Electronics Engineers, Inc,”802.1Qcc: Stream Reservation Protocol (SRP) Enhancements and Performance Improvements,” [Online]. Available: https://standards.ieee.org/standard/802_1Qcc-2018.html

[15] S. Singh, “Routing Algorithms for Time Sensitive Networks,” M.S. thesis, University of Stuttgart, 2017.

[16] M. Gutiérrez, W. Steiner, R. Dobrin, and S. Punnekkat, “Synchronization quality of IEEE 802.1AS in large-scale industrial automation networks,” Proc. IEEE Real-Time Embed. Technol. Appl. Symp. RTAS, pp. 273–282, 2017.

[17] S. S. Craciunas, R. Serna, O. Martin, and C. W. Steiner, “Scheduling real-time communication in IEEE 802.1Qbv time sensitive networks,” ACM Int. Conf. Proceeding Ser., vol. 19-21-Octo, pp. 183–192, 2016.

[18] R. S. Oliver, S. S. Craciunas, and W. Steiner, “IEEE 802.1Qbv gate control list synthesis using array theory encoding,” Proc. IEEE Real-Time Embed. Technol. Appl. Symp. RTAS, pp. 13–24, 2018.

[19] F. Dürr and N. G. Nayak, “No-wait packet scheduling for IEEE time-sensitive networks (TSN),” ACM Int. Conf. Proceeding Ser., vol. 19-21-Octo, pp. 203–212, 2016.

[20] H. Lee, J. Lee, C. Park, and S. Park, “Time-aware preemption to enhance the performance of Audio/Video Bridging (AVB) in IEEE 802.1 TSN,” 2016 1st IEEE Int. Conf. Comput. Commun. Internet, ICCCI 2016, pp. 80–84, 2016.

[21] T. Park, S. Samii, and K. G. Shin, “Design Optimization of Frame Preemption in Real-Time Switched Ethernet,” Proc. 2019 Des. Autom. Test Eur. Conf. Exhib. DATE 2019, pp. 420–425, 2019.

[22] D. Bruckner et al., “An Introduction to OPC UA TSN for Industrial Communication Systems,” Proc. IEEE, vol. 107, no. 6, pp. 1121–1131, 2019.

[23] P. Pop, M. L. Raagaard, M. Gutierrez, and W. Steiner, “Enabling Fog Computing for Industrial Automation Through Time-Sensitive Networking (TSN),” IEEE Commun. Stand. Mag., vol. 2, no. 2, pp. 55–61, 2018.

[24] L. Lo Bello, R. Mariani, S. Mubeen, and S. Saponara, “Recent Advances and Trends in On-Board Embedded and Networked Automotive Systems,” IEEE Trans. Ind. Informatics, vol. 15, no. 2, pp. 1038–1051, 2019.