

Research on distributed real-time synchronous motion control based on time-sensitive networking

Li Han*, Hong Yang, Lan Zhuo, and Chi Zhang

China Electronics Standardization Institute, China

Abstract. As a set of protocol standards defining time sensitive mechanism of Ethernet data transmission, Time sensitive network (TSN) can support transmission of isochronous data and non-periodic data in the same network. Distributed synchronous control plays an important role in large precision manufacturing scene. In this paper, a distributed real-time synchronous motion control scheme of two motors based on TSN is proposed, and the experiment shows that the two motors moving at high speed can still maintain phase synchronization under the scheduling mechanism of IEEE Std 802.1Qbv even if there is a large traffic load in the network.

1 Introduction

When the traffics from different data sources are converged and forwarded on the same switching device or routing device, they must be queued in the queues of the corresponding egress ports in the traditional Ethernet. The size of traffic in the network determines the queuing length and the queuing length determines the waiting time. If there is too much traffic in the network, there will be congestion and the queuing time cannot be predicted, so the traditional Ethernet lacks determinacy [1]. As a set of highly deterministic communication standards proposed by IEEE802 TSN working group, Time-sensitive Networking (TSN) propose a variety of mechanisms to improve real-time transmission performance of Ethernet [2], including multiple sub-protocols such as 802.1Qbv, 802.1Qcc, 802.1Qbu, 802.1Qch, 802.1Qca and etc. TSN can not only meet the functional requirements of traditional real-time Ethernet, but also provide deterministic transmission capability on the basis of Ethernet to guarantee data transmission features such as low latency and low jitter [3].

Distributed synchronous motion control is widely used in high-precision and multi-variable control systems such as textile, papermaking and steel rolling. Control performance directly affects the reliability and product quality of the whole system. Multi-motor synchronous control includes mechanical synchronous and electrical synchronous. Compared with mechanical synchronous method, electrical synchronous has better flexibility and higher control accuracy [4]. Real-time network protocol uses different

* Corresponding author: hanli@cesi.cn

methods to synchronize slave nodes with host. One common method is to configure a local synchronization clock at each node. The consensus on time ensures that command values and commands are closely synchronized across all servo axes. Real-time deterministic Ethernet protocols (such as EtherCAT) have been able to support synchronous operation of multi-axis motion control systems. But these synchronization methods cannot apply to distributed systems and the maximum bandwidth is only 100Mbit/s. These protocols are all closed and incompatible. TSN can realize high precision synchronous transmission of cooperative signals between controllers.

In [5], Moxa company developed TSN-related switches and communication equipments and realized internal network interconnection between workshops. They showed the stability of TSN technology in industrial application and promoted the development of TSN technology in the manufacturing field. In [6], a TSN integrated environment simulator was developed to realize the main functions of the TSN standards under development, and verified that the traffic used for autonomous driving could meet TSN transmission requirements in In-Vehicle Networking (VIN) and reduced the total end-to-end latency. [7] aimed at the latency and jitter problem of time-triggered cycle frame described in IEEE Std 802.1Qbv, and developed a new experimental device which achieved the lowest latency and jitter of time-triggered cycle frame, not affected by the flow load and its priority. A new network simulator NeSTiNg used for IEEE-TSN network was developed in [8]. The time gating mechanism (IEEE Std 802.1Qbv), frame preemption (IEEE Std 802.1Qbu) and shaper based on credit (IEEE Std 802.1Qav) were integrated into the simulation framework. The end-to-end latency of time-trigger flow was compared under the three different switch configurations. An OMNeT++ TSN simulation model based on CoRE4INET framework was proposed in [9], which added traffic scheduling of IEEE Std 802.1Qbv and time synchronization of IEEE Std 802.1AS.

The paper proposes a new distributed real-time synchronous motion control scheme based on TSN technology. Under the scheduling of the prototypical 802.1Qbv switches, the two motors moving at high speed can still keep phase synchronization even if there is a large flow load in the network.

The remainder of the paper is structured as follows. First, we describe the experimental schemes in section 2, then we describe the experimental process and results in detail in Section 3. The experiment results are analyzed in Section 4 and we conclude in Section 5.

2 Experimental schemes

There is the periodic real-time motor position control data frame and other non-real-time traffic competing for bandwidth in the experimental TSN network. Since IEEE 802.1Q does not have a time-triggered scheduling mechanism, high-priority frames will be forwarded first. The experimental schemes are as follows:

Scheme 1: IEEE 802.1Q (disable TSN)

Under normal conditions, the egress port of the switch forwards first data frames with high-priority and most of the bandwidth will be occupied if the traffic participating in the competition is very large. The average value and maximum value of the latency of periodic real-time traffic depend on the priority and rate of competing traffic. When the priority of competing traffic is higher than the priority of periodic real-time traffic, the average latency and maximum latency of periodic real-time traffic is greater under the greater rate of participating traffic and the greater network load. The priority of competing traffic is smaller or the same as the priority of periodic real-time traffic, and the average latency of periodic real-time traffic will also increase if the network load increases. So periodic real-time streams will have larger latency and jitter when the network reaches congestion under this scheme.

Scheme 2: IEEE 802.1Qbv and IEEE 1588 (long time-sensitive stream time slot)

TSN aims to realize the coexistence of time-critical data streams and general data streams in a network, which requires a good time scheduling function of switch equipments in the TSN network to ensure low latency and low jitter for time-critical data frames transmission. IEEE 802.1Qbv standard defines the specific implementation standards for traffic scheduling characteristics triggered by time[10]. 802.1Qbv defines a time-based shaper function allowing time-triggered communication at the egress port. The time-aware shaper is essentially a gate, and the state of the gate is uniformly scheduled and managed by the time schedule, and the transmission of queue frames is enabled or disabled according to the periodic schedule [11].

Under this scheme, the egress port of switch enabling 802.1Qbv mechanism will be open or close the transmission gate of the corresponding queue according to the configured schedule. As long as a sufficiently long time slot is given to the real-time stream within a cycle, it can be guaranteed that time-sensitive frames can be forwarded during this time slot. Synchronizing the PTP clock between the switch clock and the terminal clock is to ensure the accuracy of the time-sensitive frames sending time and the time-sensitive frames will not arrive at the switch too early or too late. Even if the priority of the load traffic is higher than the priority of the real-time stream and the network is congested, the latency and jitter of the real-time streams are still very small, and the synchronization effect of the high-speed dual motor is also the best.

Scheme 3: IEEE 802.1Qbv and IEEE 1588 (short time-sensitive stream time slot)

Compared with scheme 2, this scheme only shortens the time slot of time-sensitive streams. If the switch does not implement the frame preemption mechanism, it will be very likely to occupy the time slot when the size of the load data frame is very large. Then the real-time streams can only be forwarded in the next cycle, so it will cause the maximum latency.

3 Experiment

This research is also applicable to other industrial applications. Two servo motors that realize synchronous motion through an Ethernet switch can represent applications with periodic real-time requirements in the industry, and it needs to ensure the lowest latency and jitter. For the best-effort applications in industry, the load traffic is not very demanding for time but the bandwidth requirement is very high.

The object of this research is the latency and jitter of the real-time frame generated by the servo motor control terminal. The transmission period of the frame is 1ms, and the size including the vlan tag is 246 bytes, and the Qos priority is 5.



Fig. 1. Experimental setup.

As shown in Figure 1, the terminal on the left sends periodic and real-time frames containing position control information to the terminal on the right. Both terminals use the open source IgH EtherCAT protocol stack as the EtherCAT master stations which communicate with the Maxsine servo drive as the EtherCAT slave station to control the movement of the servo motor. Both terminals are common industrial computers (CPU: intel Four Core i7-6700 340GHz, 4G RAM, Intel I210 NIC) deployed with ubuntu 16.04 operate system (kernel: linux 5.0.21-rt16). The switch between the two terminals is an RSPE37 industrial switch with prototypical FPGA-based Ethernet ports (1Gbit/s) that supports the basic functions of IEEE 1588v2 and IEEE 802.1Qbv. The industrial computer nearby uses Qstinato traffic generator software to generate different size traffic loads and accesses into the switch.

The method of measuring latency in this paper is to measure time interval through the terminal. The sending terminal gets the current hardware time as its sending time t_{send} while sending the frame, and writes the timestamp into the frame. The receiving terminal gets the current time when the frame is received. The hardware time is regarded as its receiving time $t_{receive}$, and its latency $t_{latency} = t_{receive} - t_{send}$. The jitter is the difference between two adjacent latencies and the phase difference between two servo motors is the difference in encoder position within one cycle.

The experiment of Scheme 1: IEEE 802.1Q (disable TSN)

This experiment is mainly used to measure the effect of the priority and rate of network load traffic on the latency of real-time traffic. According to experiment scheme 1, the Qos priority of the load flow is set to 4 and 6, and the frame length is 600Bytes. During the experiment, there are different data rates of network load traffic (200 Mbit/s, 400 Mbit/s, 600 Mbit/s, 800Mbit/s, 1000Mbit/s and 1200Mbit/s) used for comparison. Sampling 60,000 latency data are used as a sample to calculate the average latency value, the maximum latency value, the minimum latency value, the standard deviation latency value, the jitter peak-to-peak value and other statistical data.

The experimental results are shown in Figure 2. Increasing the network traffic load will increase the maximum and average latency values of real-time frames. When the priority of network load traffic is greater than the priority of real-time traffic and the data rate of the network load traffic exceeds 800 Mbit/s, the average latency increases by an order of magnitude. Especially when the network becomes congested (6, 1200Mbit/s), the impact is maximum and the latency reaches 18s. At this time the motors are not synchronized and it take about 18 minutes to sample 60 latency data points. If the priority of network load traffic is less than the priority of real-time traffic and the network is congested (4,

1200Mbit/s), the latency increases sharply. Therefore, under the scheduling of IEEE 802.1Q, the latency of real-time traffic depends on the size of the network traffic load and the priority of the load traffic, and the higher priority has a greater impact.

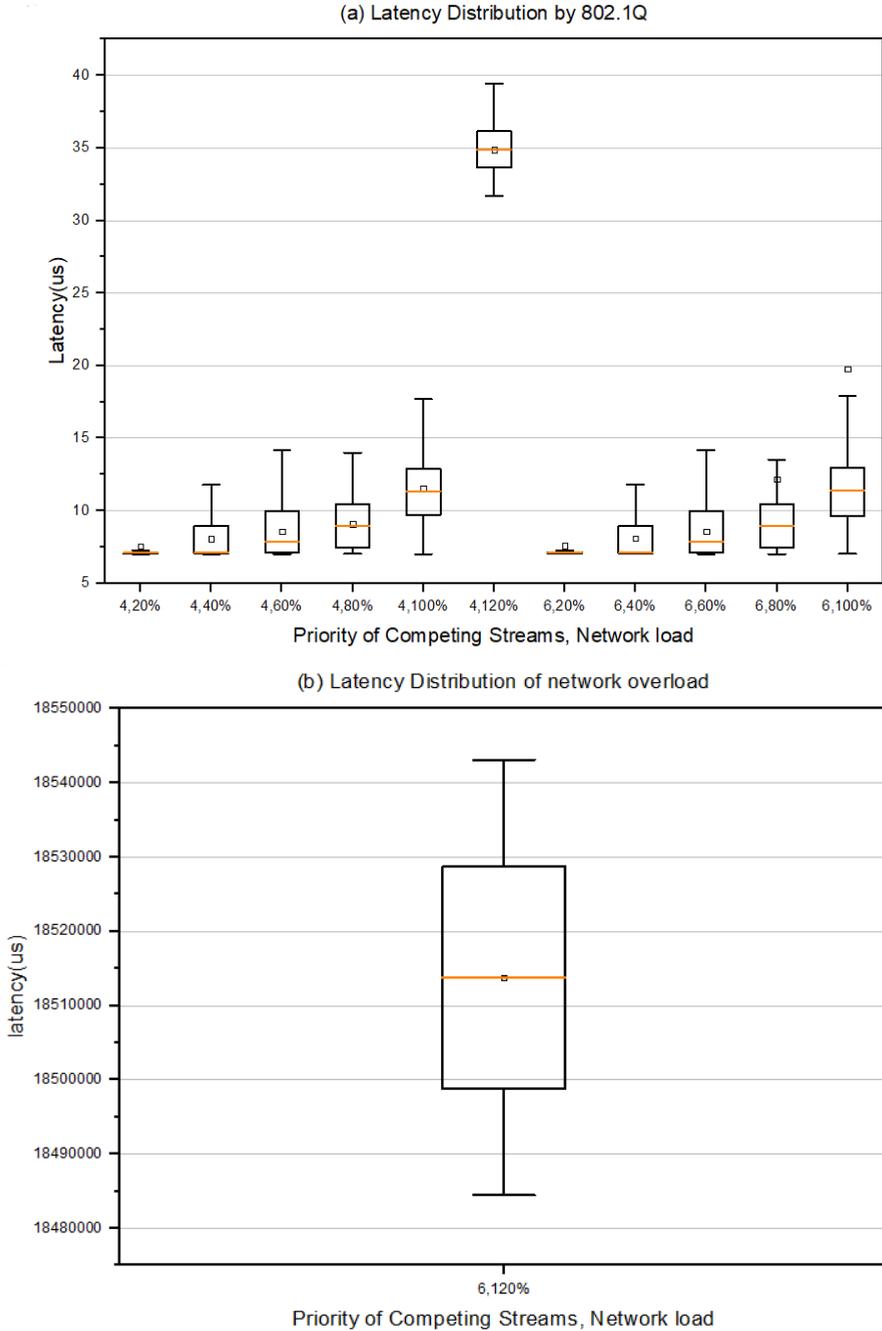


Fig. 2. Under 802.1Q, higher load and priority of non-real-time streams leads to higher latency and jitter of real-time streams.

The experiment of Scheme 2: IEEE 802.1Qbv and IEEE 1588 (time slot for real-time streams is 300us)

This experiment is mainly to verify the gate controlling feature of IEEE 802.1Qbv. GCL is used to configure schedules containing one or more time slots. Each time slot defines when and how long the gates of the corresponding QoS priority queues in an egress port are open. As in Experiment 1, the frame length of the load streams is 600Bytes, and the number of samples is 60,000. The priority of the network load traffic is set to 4 and 6 to measure the latency of the real-time frame.

In this experiment, the cycle period of GCL is set to 1000us which is equal to the period of the real-time frame. We allocate the first 400us of the cycle for the transmission of non-real-time frame, the middle 300us of the cycle for the transmission of the real-time frame exclusively, and the remaining time slot for the transmission of non-real-time frame. Based on IEEE 802.1Qbv, the sending terminal must accurately calculate the transmission time t_{send} of periodic real-time frames. In this experiment, the ETF qdisc mechanism and the intel I210 network card are used to accurately control the transmission time of each real-time frame, so as to ensure that the real-time frames are in a specific time slot when they arrive at the egress port of the switch. To calculate the transmission time t_{send} of the frame, it requires information about the time base t_{base} , the GCL cycle period t_{cycle} and its own Qbv offset time t_{offset} . t_{base} is the same for all network nodes to calculate the starting time of the next GCL cycle and t_{offset} describes the exact the designated reserved time slot position in the GCL cycle for the real-time frame. In addition, the sending terminal has to know the one-way transmission latency $t_{one-way}$ to the port of the first switch in the network. With the $t_{one-way}$, the sending terminal can send the real-time frame earlier so that the frame arrives at the switch on time. This value can be obtained by PTP clock synchronization. Based on the current system time $t_{current}$, the sending time t_{send} of the next cycle is:

$$t_{send} = \left\lceil \frac{t_{current} - t_{base}}{t_{cycle}} + 1 \right\rceil * t_{cycle} + t_{offset} - t_{one-way} \quad (1)$$

In the experiment 2, the real-time streams are transmitted through UDP and the sending time is calculated in user space, so the system time of the sending terminal must be synchronized with the hardware time.

The experimental results show that increasing the network traffic load has no effect on the latency of the real-time frames in Figure 3. No matter the priority of the load traffic is higher or lower than the priority of the real-time traffic, the average latency does not change much. Even if there is network congestion (6,1200Mbit/s), the latency and jitter are very small. Compared with the results of Experiment 1, the lowest latency and jitter are achieved. Therefore, under the scheduling of 802.1Qbv, the latency of the real-time frame is independent of the priority and rate of network traffic and the lowest latency and jitter are realized. The position difference of the dual motors is also the lowest.

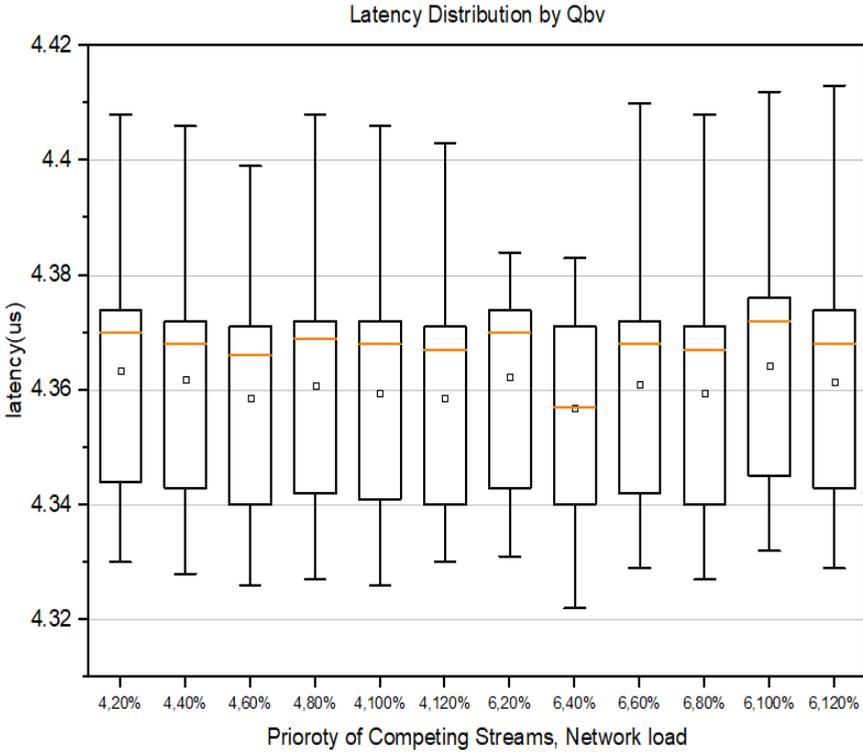


Fig. 3. 802.1Qbv achieves the lowest latency and jitter for real-time streams.

In addition, 802.1Qbv can avoid packet loss. Due to ETF qdisc timeout discarding mechanism, 60,000 data packets are sent from user space but there may not be 60,000 data packets actually sent from the I210 network card. The Beckhoff ET2000 is connected between the sending terminal and the switch to measure the number of data packets flowing into the switch and the number of data packets flowing out of the switch is measured in the receiving terminal in the experiment to test packet loss. In this experiment, the switch can be disable TSN or enable TSN, the priority of load traffic is 6 and the frame length is 1500Bytes. The experimental results are shown in Table 1. When TSN is disabled and the network traffic reaches 100% and 120%, the switch begins to lose packets, especially when it is overloaded (120%), only 2% packets pass the switch. When TSN is enabled, even if the network traffic reaches 120%, there is no packet loss.

Table 1. 802.1Qbv can avoid packet loss.

category	NO TSN, 20%	NO TSN, 40%	NO TSN, 100%	NO TSN, 120%	TSN, 100%	TSN, 120%
Sent packets	59536	59519	59399	55270	59036	59001
Received packets	59536	59519	58855	1217	59036	59001
Lost packets	0	0	544	54053	0	0

The experiment of Scheme 3:IEEE 802.1Qbv and IEEE 1588 (time slot for real-time streams is 10us)

This experiment is mainly to measure the maximum latency by IEEE 802.1Qbv. Compared with experiment 2, the time slot for real-time streams in the GCL schedule of the switch is shortened. When the load flow frame is very long and the switch does not implement the frame preemption mechanism, the load frame may occupy the time slot that should be used to forward the real-time frame, which leads to real-time frame to continue to be stored in the queue and forwarded only in the next cycle. There is the maximum latency. In this experiment, the priority of the load traffic is 6, the frame length is 1500Bytes and the load rate is 1200Mbit/s. For 1Gbit/s Ethernet ports, a 1500-byte Ethernet frame needs 12us to be sent, so the time slot used for real-time streams is set to 10us.

The experimental results are shown in Figure 4. In the case of network congestion (6, 120%), the maximum latency of real-time streams is about 1000us of the cycle period of GCL, which shows that the time slot used for forwarding the real-time frames is occupied by load frames. But even in this worst case, the maximum delay can be guaranteed and the high-precision synchronization of motors is also guaranteed.

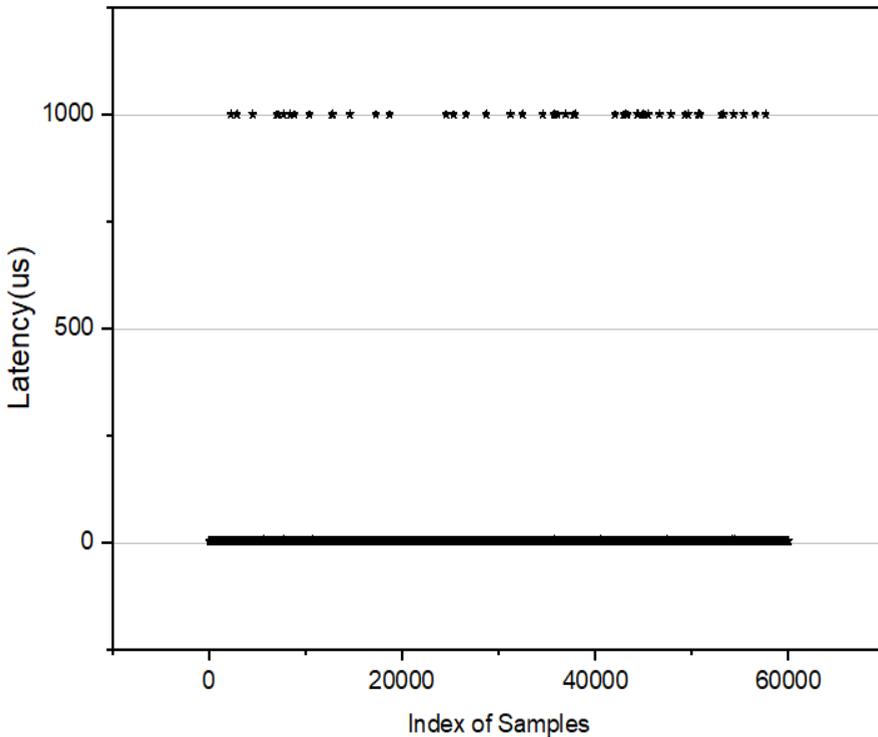


Fig. 4. Under 802.1Qbv, the maximum latency can be guaranteed even if the network is congested.

4 Discussion

Considering the results of the above experiments, IEEE 802.1Q cannot guarantee the maximum latency. Network congestion leads to uncertain delay so it is suitable for time-insensitive applications, such as audio and video streaming. For applications that require low latency and low jitter, such as motion control, IEEE 802.1Qbv can be applied. As long as the real-time stream is given a sufficiently long time slot to ensure that the real-time frame can be forwarded in the time slot of this cycle, the lowest delay and jitter can be

achieved. Even in the worst case of the network, the maximum latency equal to the GCL cycle period can be guaranteed.

5 Conclusion

We presented a new distributed real-time synchronous motion control scheme based on TSN technology. The experimental results show that as long as the time slot of real-time stream is sufficiently long for the switch enabling IEEE 802.1Qbv and PTP, the minimum latency and jitter can be achieved even if the frame preemption mechanism is not implemented. In this case, the latency is independent of the priority and rate of network traffic and the high-speed dual motors can maintain better phase synchronization. If the time slot reserved for the real-time stream is relatively short, it can also ensure that the maximum latency is within a boundary, which also verifies the deterministic behavior of TSN.

There are still many deficiencies in this study. For example, the method of measuring latency is not very accurate and there is a large error when measured by the terminal. In a future work, we plan to develop FPGA-based latency measurement devices and IEEE 802.1Qbv configuration tools.

References

1. Xiaolin Xu. Time-sensitive Networking Technology and its Application in Industrial Internet[J]. Telecommunications Network Technology,2018(05):1-5.
2. Peng Li. 10G-based Deterministic Network Node Card Logic Design[D].University of Electronic Science and Technology of China,2018.
3. IEEE 802.1 Time-Sensitive Networking (TSN) Task Group. [Online]. Available: <https://1.ieee802.org/tsn>.
4. PEREA-PINAL F J, NUNEZ C, ALVAREZ R, et al. Comparison of Multi-motor Synchronization Techniques[C]. 30th Annual Conference of the IEEE Industrial Electronics Society. Busan :IEEE, 2004:1670-1675.
5. Hengyu Wang. Moxa Passed the Network Devices Interoperability Test of Alliance of Industrial Internet [J]. Computer & Network, 2019,45(11):73.
6. Lee Juho, Park Sungkwon. Time-Sensitive Network (TSN) Experiment in Sensor-Based Integrated Environment for Autonomous Driving[J]. Sensors (Basel, Switzerland), 2019,19(5).
7. M. H. Farzaneh, and A. Knoll. “Time-Sensitive Networking (TSN): An Experimental Setup,” in IEEE Vehicular Networking Conference (VNC), 2017, pp. 23-26.
8. J. Falk, D. Hellmanns, and B Carabelli. “NeSTiNg: Simulating IEEE Time-sensitive Networking (TSN) in OMNeT++,” in International Conference on Networked Systems (NetSys), 2019.
9. J. Jiang, Y. Li, S. H. Hong, A. Xu, and K. Wang. “A time-sensitive networking (TSN) simulation model based on OMNeT++ ” . in IEEE Int. Conf. Mechatronics and Automation (ICMA), Aug. 2018, pp. 643 - 648.
10. IEEE 802.1Qbv-Enhancements for Scheduled Traffic. [Online]. Available: <http://www.ieee802.org/1/pages/802.1bv.html>.

11. Silviu S. Craciunas, Ramon Serna Oliver, Martin Chmelík, and Wilfried Steiner.
“ Scheduling Real-Time Communication in IEEE 802.1Qbv Time Sensitive Networks,” in Proc. RTNS(2016), ACM.