QoS Comparison of ATS and CBS in IEEE 802.1TSN over In-Vehicle Ethernet Based on Automotive Use Case

Akari Yoshimura and Yoshihiro Ito
Graduate School of Engineering Nagoya Institute of Technology
Nagoya, 466-8555, Japan
Email: {yosimura@en., yoshi@}nitech.ac.jp

***Abstract*—This paper studies the QoS of Asynchronous Traffic Shaping (ATS), one of the QoS controls defined in IEEE 802.1TSN, over in-vehicle Ethernet configured based on the use cases in IEEE P802.1DG. This study evaluates QoS by experiment and compares it with Credit-Based Shaper (CBS), one of the competing QoS controls defined in IEEE 802.1TSN. The experiment uses traffic whose specification is defined in the above-mentioned use case; it treats the maximum delay, the variance of delay, and the frame loss rate as QoS parameters to be evaluated. The experimental results indicate that as the amount of traffic increases, ATS can reduce the maximum delay and the variance of delay, but the frame loss rate increases instead. On the other hand, the results for CBS show that CBS can reduce the frame loss rate. Therefore, if the requirement for delay is more important than that for frame losses, it is better to adopt ATS. Otherwise, CBS is appropriate to adopt.**

I. Introduction

Using high-speed Ethernet is essential for the realization of fully autonomous vehicles because it transmits large amounts of data over its in-vehicle network. On the other hand, since various data are transmitted over in-vehicle Ethernet, losses or delays of safety-relevant data can occur at some failure of the vehicle. Therefore, adopting some QoS control to guarantee low latency and low frame loss rate of safety-relevant data over the Ethernet is necessary. Thus, IEEE 802.1TSN (Time-Sensitive Networking) standard[1] considers a QoS control.

IEEE 802.1TSN standard consists of many standards and defines QoS controls, such as Strict Priority Queueing (SPQ), Credit Based Shaper (CBS), Time-Aware Shaper (TAS), and Asynchronous Traffic Shaping (ATS) in the IEEE 802.1Q[2], which is one of IEEE 802.1TSN. SPQ is the most basic QoS control, which transmits frames according to the priority of the frames. CBS enables more flexible QoS control than SPQ by adding a variable called Credit to limit the maximum bandwidth usage of traffic on each priority. TAS performs QoS control more accurately than SPQ and CBS by opening and closing gates on the egress port according to a predefined schedule. However, TAS requires time synchronization between devices, such as switches. On the other hand, ATS does not require time synchronization and is expected to provide lower jitter and delay than the other standards at a low implementation cost.

Although QoS evaluations of SPQ, CBS, and TAS are performed in [3]-[6], studies on evaluating QoS of ATS are few since ATS is a newer technique than the others. Therefore, it is inevitable to evaluate the QoS of ATS. Especially as CBS is a competing control of ATS, a comparison between ATS and CBS is required.

This paper evaluates QoS provided by ATS and CBS over an experimental network constructed based on the use cases in IEEE P802.1DG by simulation. The rest of this paper is organized as follows. Section II introduces an overview of the ATS. Sections III and IV show the experiments and the results, respectively. Finally, we conclude this paper in Sect. V.

II. Asynchronous Traffic Shaping (ATS)

Asynchronous Traffic Shaping (ATS) is one of the QoS controls defined in IEEE802.1Q. Figure 1 shows an overview of ATS. In Fig. 1, each switch receives time information from its independent clock and transmits frames based on a scheduler called a shaper, which implements the ATS algorithm attached to the queue. When the amount of input traffic exceeds the expected traffic amount designated in a traffic specification, the shaper delays excess traffic to equalize the amount of traffic by keeping frames in the queue.

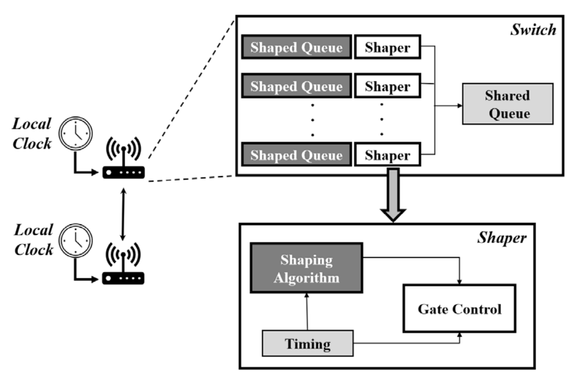


Fig. 1: Asynchronous Traffic Shaping

The scheduler controls based on the Token Bucket Emulation (TBE) scheme to work asynchronously. Figure 2 introduces TBE. As shown in Fig. 2, transmission rights called tokens are continuously placed into a buffer called a token bucket at a fixed rate. Since the token bucket has a limited capacity, the newly arriving tokens are discarded when the token bucket is full. If there need to be more tokens in the token bucket to transmit a frame, the frame will either be held for transmission until enough tokens are accumulated in the token bucket, it will be discarded, or sent with a lower priority.

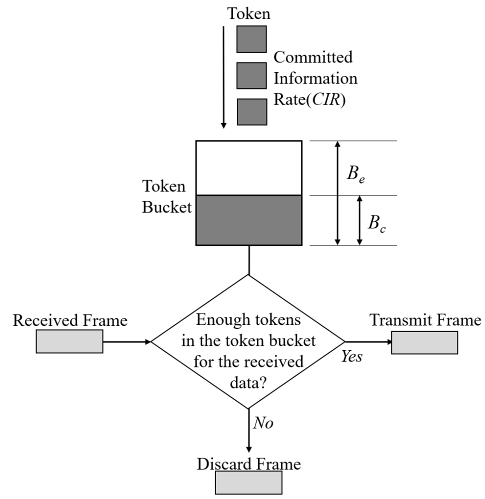


Fig. 2: Token Bucket Emulation

ATS has some configurable parameters as shown in Fig. 2. *CIR*(Committed Information Rate) is the input rate of the tokens, *Be*(Burst Exceed) means the maximum number of tokens that can be stored in the token bucket, and *Bc*(Burst Committed) indicates the number of tokens can be removed at one time. Consequently, frame delay can be limited in ATS without time synchronization between devices. As above-mentioned, since ATS does not rely on time synchronization, it is expected to reduce implementation costs.

III. Experiments

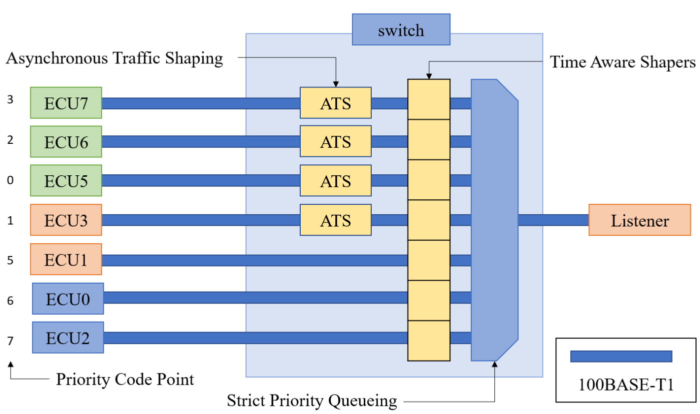


Fig. 3: Experimental network

Figure 3 shows the experimental network. This network is based on the use case discussed in IEEE P802.1DG and consists of seven ECUs(Electronic Control Units) for frame transmission (ECU0 through ECU7), one switch (Switch), and one ECU for frame reception (Listener). Each frame to be transmitted is assigned one of seven priorities. Although ECU4 with a priority of 4 initially applies Frame Preemption (FP) in [7], this paper does not consider FP and omits ECU4. SPQ and TAS are used to the egress port of switches for all priorities, and ATS is applied to priorities 0 through 3. The specifications of ECUs are shown in Table I. Here, the requirement for the maximum delay is defined based on the previous study [8].

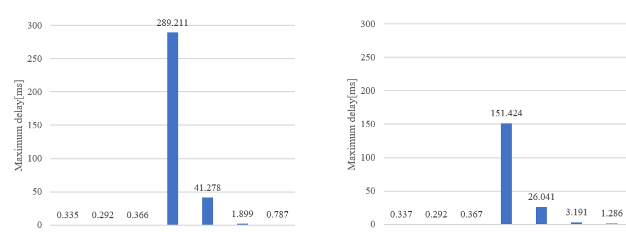
TABLE I: Specification of experimental traffic

ECU	PCP	Interval[ms]	Frame Size[byte]	Required Value[ms]
ECU7	3	0.125 - 1.5	1470	3
ECU6	2	0.125 - 1.5	1470	5
ECU5	0	0.125 - 1.5	1470	50
ECU3	1	0.125 - 1.5	1470	300
ECU2	7	0.25	1470	None
ECU1	5	10	64	1
ECU0	6	1	512	None

In this experiment, the transmission interval of frames with priorities 0 through 3 varies between 0.125 and 1.5 milliseconds to increase the transmission instantaneously. Our experiments are conducted where the ATS does not control the transmission and controls the transmission; they will also be done with CBS instead of ATS to compare QoS between ATS and CBS. The value of the GCL of TAS is set so that the frame transmitted by ECU1 is not affected by the other traffic. The value of ATS's *CIR* is controlled as 10 Mbit/s for all traffic to which the ATS is applied to be uniform, and the maximum delay to be set is the value according to the requirement. Our experiments are done by simulation and use OMNeT++[9] as our simulation platform.

IV. Results

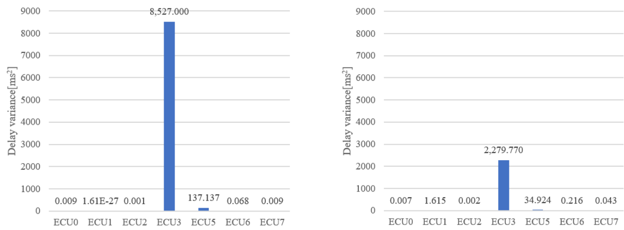
The experimental results are shown in Figs. 4 through 6. The abscissa represents the ECU, while the ordinates in Figs. 4, 5, and 6 illustrate the maximum delay, the variance of delay, and the frame loss rate, respectively. From Fig. 4, we see that the maximum delay of low-priority traffic is suppressed when ATS is applied since frames can only be transmitted within the bandwidth determined by the value of *CIR*. Because of the maximum delay configuration, all cases where ATS had been applied resulted in the requirement for the maximum delay being met. Also, Fig. 5 indicates that the traffic applied ATS has less variance of delay because of the control for the maximum delay by ATS, and all the variance of delay can be suppressed compared to the case without ATS applied. Although Fig. 6 shows that the frame loss rate of ECU3 is larger when ATS is not applied, the traffic of ECU3 can be transmitted with the same frame loss rate using ATS.



(a) No ATS applied (b) ATS applied

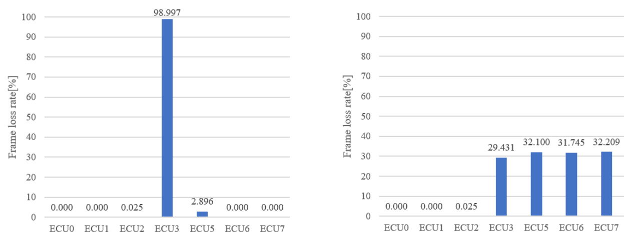
Fig. 4: Maximum delay with/without ATS

Next, the results of applying CBS instead of ATS are shown in Figs. 7 and 8. For comparison, the ATS data presented



(a) No ATS applied (b) ATS applied

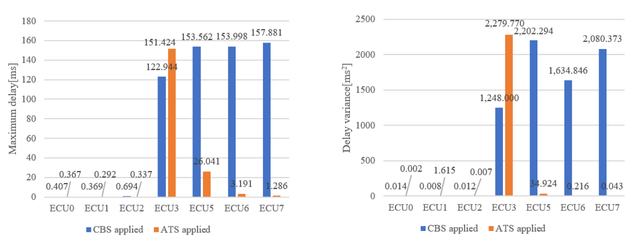
Fig. 5: Variance of delay with/without ATS



(a) No ATS applied (b) ATS applied

Fig. 6: Frame loss rate with/without ATS

before is also shown. The abscissa of the figures represents ECUs, and the ordinates of Figs. 7 and 8 mean both the maximum delay and the variance of delay, and the frame loss rate, respectively. Figure 7 shows that the ATS results are better for all but ECU3, which has the lowest priority. This is because the maximum delay and the variance of delay are suppressed due to the limitation of the delay. Conversely, from Fig. 8, we see that the frame loss rate of low-priority traffic for CBS is better for CBS than for ATS. This is because the ATS controls to meet the requirement for the maximum delay configuration.



(a) Maximum delay (b) Variance of delay

Fig. 7: Maximum delay and variance of delay for CBS

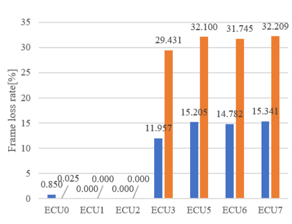


Fig. 8: Frame loss rate for CBS

On the other hand, since *idleSlope* in CBS allows bandwidth to be reserved, frame losses can be reduced.

The abovementioned results indicate that implementing ATS is suitable for suppressing maximum delay and variance of delay even for low-priority frames when the amount of transmission data increases. However, the maximum delay and variance of delay are suppressed while the frame loss rate for low-priority traffic increases. Compared to CBS, the maximum delay and the variance of delay will be suppressed when ATS is applied, while the frame loss rate will be suppressed more when CBS is used. To summarize, ATS must be used when the requirement for the maximum delay or the variance of delay is dominant, while CBS is appropriate when the requirement for the frame loss rate is dominant.

V. Conclusions

This paper studied the QoS of ATS, one of the IEEE 802.1TSN standards, through simulation-based experiments while comparing the QoS of CBS. From the experimental results, we found that ATS can suppress the maximum delay and the variance of delay for low-priority traffic at congestion. However, suppressing the frame loss rate for low-priority traffic increases the frame loss rate for the other traffic. On the other hand, although ATS suppresses the maximum delay and the variance of delay, the frame loss rate becomes worse than CBS. As a result, we conclude that ATS is appropriate to meet the requirement for the maximum delay or the variance of delay over in-vehicle networks. At the same time, CBS is appropriate to meet the requirement for the frame loss rate.

Our future works are as follows. First, we would like to tackle the other in-vehicle networks based on different use cases. Second, we will treat various types of traffics for evaluation. Third, we will consider frame preemption.

References

[1] "IEEE 802.1 Time-Sensitive Networking (TSN) Task Group," https://1.ieee802.org/tsn/

[2] "IEEE 802.1Q-2018 IEEE Standard for Local and Metropolitan Area Networks–Bridges and Bridged Networks," https://standards.ieee.org/ieee/802.1Q/6844/

[3] K. Hirano and Y. Ito, "Study on Appropriate IdleSlope Value of Credit Based Shaper for QoS Control on In-Vehicle Ethernet," Proc. the 9th Global Conference on Consumer Electronics (GCCE), Oct. 2020.

[4] C. Kobayashi and Y. Ito, "Study on Effect of Time Synchronization Precision on QoS with Time-Aware Shaper," Proc. the 10th Global Conference on Consumer Electronics (GCCE), Oct. 2021.

[5] M. Kim, D. Hyeon and J. Peak, "eTAS: Enhanced Time-Aware Shaper for Supporting Nonisochronous Emergency Traffic in Time-Sensitive Networks," IEEE Internet of Things Journal, Volume: 9, pp. 10480 - 10491, 2021.

[6] M. Máté, C. Simon and M. Maliosz, "Asynchronous Time-Aware Shaper for Time-Sensitive Networking," Proc. the 17th International Conference on Network and Service Management (CNSM), Oct. 2021.

[7] D. Pannell, "Choosing the Right TSN Tools to meet a Bounded Latency," IEEE 802.1 TSN Interim, Sept. 2020.

[8] Y. Ozawa and Y. Ito, "Evaluation of TCP Performance Under In-Vehicle Networks With IEEE 802.1 TSN," Proc. the 11th Global Conference on Consumer Electronics (GCCE), Oct. 2022.

[9] "OMNeT++ Discrete Event Simulator," https://omnetpp.org/