Empirical Evaluation of Traffic Shaping Algorithms for Time Sensitive Networking

Kunj Naik
Wireless Information Networking Group
Dept. of CSE, NITK Surathkal
Mangalore, India
kunj.203cs004@nitk.edu.in

Deepa Kumari
Wireless Information Networking Group
Dept. of CSE, NITK Surathkal
Mangalore, India
deepakumari.207cs500@nitk.edu.in

Mohit P. Tahiliani
Wireless Information Networking Group
Dept. of CSE, NITK Surathkal
Mangalore, India
tahiliani@nitk.edu.in

***Abstract*—Standard Ethernet networks cannot provide solutions to handle latency sensitive applications efficiently. The packet scheduling algorithms like First In First Out (FIFO), Class Based Queueing (CBQ), and others do not provide efficient solutions to Quality of Service (QoS) parameters like end-to-end delay, packet loss, and jitter. Time Sensitive Networking (TSN) can be used as a solution to provide QoS to time sensitive applications. TSN has emerged as a future of real-time communication. The main advantage of TSN is that it enables determinism by supporting time critical traffic while the best effort traffic is also present in the network. This paper explores two of the most popular and widespread traffic shaping mechanisms in TSN: Time Aware Shaper (TAS) and Credit Based Shaper (CBS). IEEE 802.1Qbv is used for delivering time assurance using TAS. CBS is a key traffic shaping algorithm to provide bandwidth assurance to the time critical and real time traffic, such as the audio traffic. This paper evaluates TAS and CBS using TSN enabled Network Interface Cards (NIC) with time synchronization, real time kernel and real traffic, which includes time sensitive traffic and elastic background traffic.**

***Keywords*—TSN, Time Aware Shaper, Credit Based Shaper.**

I. Introduction

Traditional Ethernet networks provide scalability and high flexibility but lack determinism and reliability. Hence, Ethernet cannot be used for highly time critical applications and real time communication, which are the core requirements of Industry 4.0. This led to the advent of Time Sensitive Networking (TSN), a set of standards that define mechanisms to provide low latency, low packet delay variation (jitter), and low packet loss for deterministic Ethernet networks. TSN Ethernet is used for time critical applications like automobile manufacturing, food processing industries, machine-to-machine/controller to controller communications and several other high precision type communications.

TSN standards are defined in IEEE 802.1 [8]. The main components of TSN include: a centralized coordinator, traffic shapers, resource manager, security provider, and latency handler. A centralized coordinator provides functionalities like time synchronization and error correction (IEEE 802.1AS), scalability (IEEE 802.1Qcc), availability (IEEE 802.1Qbu), and reliability (IEEE 802.1CB). Traffic Shaper provides functionalities like Credit Based Shaper (CBS) [1] (IEEE 802.1Qav), Time Aware Shaper (TAS) [2] (IEEE 802.1Qbv), frame preemption (IEEE 802.1Qbu), Asynchronous Traffic Shaper or ATS (IEEE 802.1Qcr), per frame filtering and policing (defined in IEEE 802.1Qci), and provides end-to-end management of traffic streams (defined in IEEE 802.1Qat). Functionalities like bandwidth allocation and stream metrics monitoring (IEEE 802.1 Qat) are provided by the resource manager. Security provider prevents bandwidth violations, malicious attacks, and DoS attacks (IEEE 802.1Qci). Latency handler provides functionalities like cyclic queuing, forwarding (IEEE 802.1Qch) and calculating worst case latency.

This paper focuses on traffic shapers (and the internal queue disciplines used therein), like CBS, TAS, ATS, and others. This work aims to evaluate different traffic shapers and compares their performance by using metrics such as end-to-end delay and jitter. Although this type of evaluation has been done in the recent past, to the best of our knowledge, it used synthetic traffic generators to generate time-sensitive traffic and non-real time kernels to perform the evaluation. The main goal of this work, instead, is to conduct the evaluation with the following three aspects: (a) using real traffic that is time sensitive in nature (for example, audio traffic). Interactive audio traffic is the most relevant type of traffic to be considered for evaluating the effectiveness of traffic shapers in a TSN environment. As audio traffic latency should be bounded, and this traffic expects lesser packet loss and jitter to get an accurate output, understanding how traffic shapers can handle them is interesting. (b) using real time kernel because it provides bounded latency and real time schedulers for the high priority and time-critical tasks, so it provides faster response time compared to the normal kernel. In a simple operating system, different critical sections are non-preemptable. The motivation to use a real-time kernel is to remove these non-preemptable critical sections and avoid disabling preemption, interrupts, locks, etc. (c) using real TSN NICs. Time synchronization is an essential functionality for the experimentation of the traffic shapers, so PTP (Precision Time Protocol) [7] provides time synchronization by synchronizing the hardware clock of time sensitive NIC.

II. Background and Related Work

TSN enabled NICs consist of the necessary hardware required to obtain a deterministic behavior by providing functionalities such as a centralized coordinator to enable time synchronization. However, only TSN NIC is not sufficient to provide determinism. Several optimizations are required in traffic shapers, which are the most important among the software components of TSN to make a deterministic network. Traffic shapers are crucial because they provide different traffic classes with different priorities to work on the same network even though each traffic class has different bandwidth and end-to-end delay requirements. They make communication efficient because, without traffic shapers, packets can arrive in bursts and overflow the buffers. The traffic shapers smooth out the traffic and distribute it evenly in time.

A. Time Aware Shaper (TAS)

TAS is the fundamental part of TSN and is defined in IEEE 802.1Qbv. TAS is designed to improve performance and efficiency by separating the fixed time slot into high priority Scheduled Traffic (ST) and low priority Best Effort Traffic (BE). This mechanism is achieved by using Gate Control Lists (GCL). GCL contains specific time windows for time-critical traffic and best effort traffic. TAS provides the information about

1) How many time slots are available per cycle?
2) Duration of each time slot.
3) The flow that can be transmitted based on the priority.

B. Credit Based Shaper (CBS)

To enhance prioritization and bandwidth optimization, TSN supports CBS as defined in IEEE 802.1Qav. This functionality provides information about the maximum bandwidth share allotted to a specific queue. CBS prioritizes high priority Scheduled Traffic (ST) over low priority Best Effort (BE) traffic. CBS allows audio data to be transferred based on class and priority. Audio, in general, has a greater priority and is classified as Class A data. On the other hand, video is classified as Class B and has a lesser priority. CBS creates credits based on the incoming class frame to avoid starvation of best-effort traffic and improve the deterministic transmission of higher-class data. CBS operates based on the credit value. The transmission of the time-critical AVB frame will be allowed when the 'credit' is zero or positive and when the channel is idle. The sender must wait until it becomes positive or zero when credit is negative.

C. Queue Disciplines

Queue discipline (qdisc) specifies the rules for handling packets at a device's incoming and outgoing queue. In traffic shaping, packets are queued when traffic exceeds the available capacity. Hence, the performance of the system is highly dependent on the queue discipline we use. Besides shaping the outgoing traffic, queue discipline can also be used to perform scheduling and policing the incoming traffic. The queue disciplines contain the following elements:

1) Queues: Stores the packets waiting for transmission.
2) Classes: Used to provide dedicated services to the flows
3) Filters: Used to segregate the flows and indicate a particular packet belongs to which class or queue.

There are two types of queue disciplines:

1) Classless: There is just one level of queuing since the classless qdisc does not include another qdisc. Classless qdisc can tell only if a packet is categorized, delayed, or discarded. Classless qdiscs can be used as a primary qdisc at the interface. It can also be used along with classful qdiscs. First In First Out (FIFO), Stochastic Fair Queuing (SFQ), Extended Stochastic Fair Queuing (ESFQ), Token Bucket Filter (TBF), and Credit Based Shaper (CBS) are the examples of the classless queue disciplines.

2) Classful: These queue disciplines support configurations using classes. As a classful qdisc can contain another qdisc, there are many levels of queues in them. Classful qdisc includes filters that direct the flows to a specified class or queue. Different filters can select from which qdisc the packets will be sent. The class attached to the root qdisc is called the root class. The terminal class attached to the qdisc is called the leaf class. Leaf class can also be a classless qdisc. Hierarchical Token Bucket (HTB), Priority Scheduler (PRIO), Class Based Queuing (CBQ), and Time Aware Priority Shaper (TAPRIO) are examples of the classful queue disciplines.

This paper focuses on two queue disciplines: the Credit Based Shaper, which uses CBS queue discipline from classless category, and the Time Aware Shaper, which uses TAPRIO queue discipline from classful category. These queue disciplines are already implemented in the mainline of the Linux kernel. The parameters that have been considered to evaluate the performance of these algorithms are per packet end-to-end delay and per packet jitter.

D. Related Work

Some related work has been done to evaluate the performance of the traffic shapers. In [6], [2] and [12], different research on optimizing the performance of 802.1Qbv (TAS) has been discussed. In [5] and [4], different performance based analysis methods are presented for the CBS. While most of the research on traffic shapers is for single traffic shapers only, some papers have researched combinations of the traffic shapers. [3] presents the combined usage of TAS and CBS for in-vehicle networks. [17] presents a timing analysis of the combined use of TAS and CBS on AVB traffic using network calculus. This study is extended to an arbitrary number of AVB flows for the same TAS and CBS architecture combination [16]. The research on the combination of TAS and CBS using simulation is done in [10].

Some papers compare different traffic shapers. In [9], the authors took six switches and six nodes and made a topology to do a simulation in OMNeT++ [14]. They also derive the network calculus-based relation and compare results with simulation results. In [15], different ring and mesh topologies are used to perform experiments on different traffic shaping algorithms using the OMNeT++ simulator. The authors also made a quantitative comparison of different traffic shapers. In [11], the paper proposed a pseudo code for a TAS and

ATS, and presented solutions like adaptive slotted window and adaptive bandwidth sharing mechanisms for TAS limitations. The paper also compares the results of ATS to find out if ATS can achieve similar results for industrial networks with periodic and sporadic data transfer. In [13], a Python based simulator (called PYTSN) is used to perform experiments on the 802.1Qbv TAS to evaluate if the TSN based TAS can be used in Tactile Cyber-Physical Systems (TCPS). It shows that TAS is optimal for the applications such as industrial processes with extremely low packet transmit intervals and bandwidth. As TCPS requires relatively higher transmission intervals and high bandwidth, TAS cannot be used as in TCPS.

III. Testbed Setup and Configurations

We use a simple talker-listener topology as shown in Fig. 1 for the experiments. The talker and listener are configured with a real time kernel. We configure realistic traffic patterns by using multi-stream audio as scheduled traffic and UDP/TCP/SCTP traffic as best effort traffic. To transmit audio traffic, we have to handle time synchronization, setup a real-time kernel, configure qdiscs and establish a VLAN interface. The following qdiscs are considered for evaluation in this paper: MQPRIO, TAPRIO, CBS with MQPRIO and CBS with TAPRIO, with end-to-end delay and jitter being the parameters for evaluation.

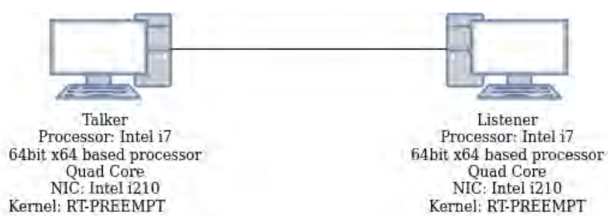


Fig. 1: TSN Testbed Setup

A. Handling Time Synchronization

This is a core feature of TSN. It is defined by the IEEE 802.1AS and is known as the Generalized Precision TIme Protocol (gPTP). gPTP is designed for time sensitive applications. It introduces the Grand Master (GM) role, which any node in the gPTP domain can play. All other nodes in the gPTP domain receive clocking information from the GM.

Linux PTP is the implementation of the gPTP in Linux. We use it to provide time synchronization as it supports the gPTP and other profiles of PTP. Linux PTP provides tools like ‘ptp4l’, ‘phc2sys’, and ‘pmc’ to configure time synchronization. *ptp4l* is a daemon that synchronizes the PTP Hardware Clock (PHC) from the NIC. *phc2sys* is a daemon that synchronizes the PHC and the system clock. *pmc* is a utility tool to configure *ptp4l* in run time.

B. Setting up a real-time kernel

Standard Linux does not support real-time capabilities. In this work, we applied an RT-PREEMPT patch on the Linux kernel to it real-time. As the Linux network subsystem is not designed for bounded latency, we use RT-PREEMPT for deterministic communications. The key advantage of the RT-PREEMPT kernel is that standard Linux tools and libraries can be used without using any real-time APIs.

C. Configuring Qdiscs

We use the traffic control (tc) utility in Linux kernel to configure qdiscs. *tc* is an important part of the Linux network system. It provides functionalities like traffic shaping, traffic scheduling, and traffic policing. The main element of traffic control is the qdisc. It enables network traffic disciplines to create queues and Quality of Service (QoS) rules for communication. Qdisc can be used to avoid traffic congestion on the transmission path. The default classless qdisc is PFIFO FAST, with three FIFO priority bands.

D. VLAN Configuration

Audio traffic is sent through VLANs. Both talker and listener are configured using a VLAN interface. The ‘ip link’ command from *iproute2* package is used to create VLAN interfaces. The ‘enp2s0’ interface, which will be used in the subsequent sections, is a TSN NIC interface.

E. Traffic Configuration

To generate audio traffic, we synchronize the hardware clock and system clock for time accuracy. However, we generate aperiodic audio traffic for our experiments to match a realistic audio traffic behavior. To generate the best effort traffic, we use *iperf3* tool. We configure it to generate the best effort traffic, which is a combination of UDP, TCP and SCTP flows. We choose these different types of flows to ensure that the best-effort traffic leads to an increase in the contention level for the audio traffic. The idea is to evaluate how different qdiscs perform in terms of achieving time-sensitive behavior when the load of best effort traffic is relatively higher.

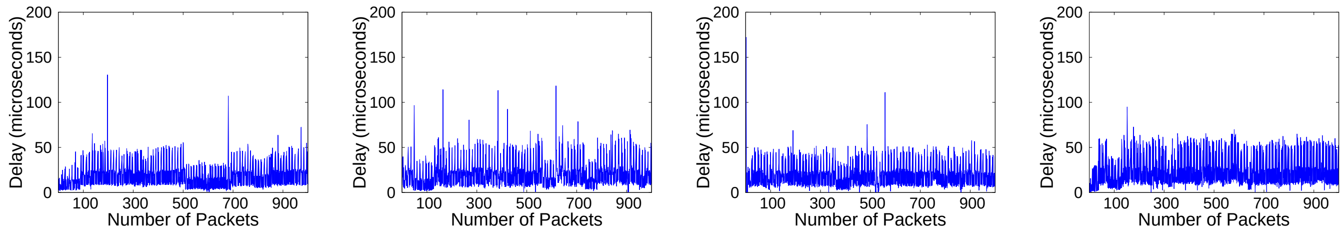
IV. Results and Analysis

We perform two sets of experiments. In the first set, we generate only audio traffic in the network and evaluate the performance of qdiscs in terms of delay and jitter. In the second set, we generate best-effort traffic alongside the audio traffic and re-evaluate the performance of qdiscs.

A. Audio traffic only

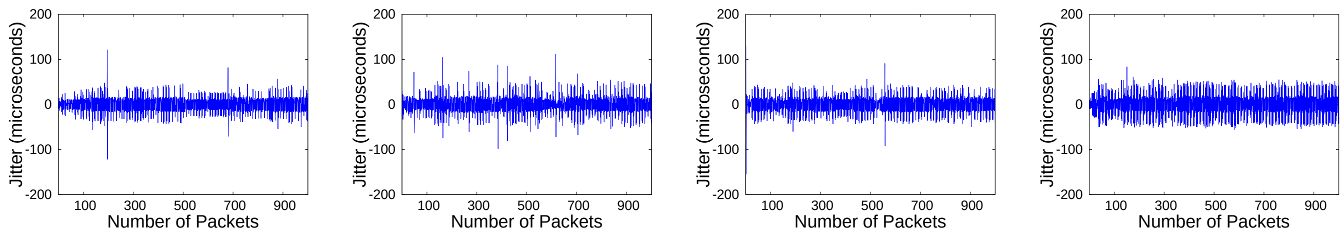
Figures 2a-2d and Figures 3a-3d depict the results obtained for per packet delay and jitter, respectively, for all the qdiscs. The average delay for the MQPRIO is 19.896 microseconds and for TAPRIO is 8.940 microseconds. MQPRIO uses the priority to map traffic flows to hardware queues. It maps traffic flows to traffic classes using priorities, and traffic classes are mapped with the hardware queues. On the other hand, TAPRIO not only maps traffic flow to hardware queues like MQPRIO, but also ensures that each frame transfer from queues is time-based. Due to time-triggered scheduling, TAPRIO has lower delay than MQPRIO.

Since CBS does not have the ability to map traffic to the hardware queues, it is installed under the classful qdisc, which maps the traffic flows to hardware queues. MQPRIO and



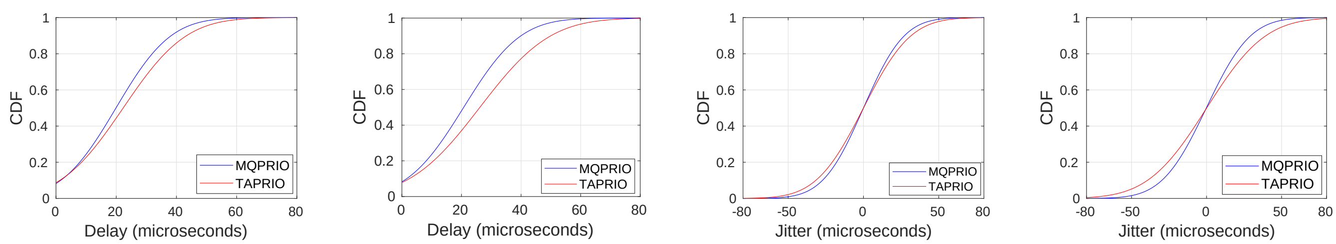
(a) MQPRIO: audio only (b) TAPRIO: audio only (c) CBS+MQPRIO: audio-only (d) CBS+TAPRIO: audio only

Fig. 2: Per Packet Delay: Audio only



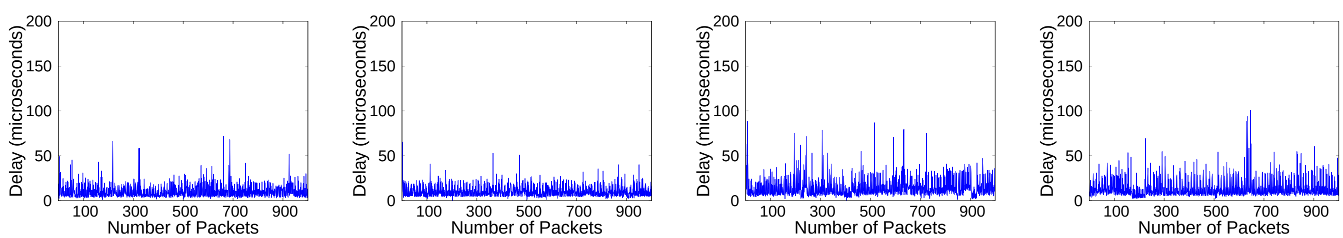
(a) MQPRIO: audio only (b) TAPRIO: audio only (c) CBS+MQPRIO: audio-only (d) CBS+TAPRIO: audio only

Fig. 3: Per Packet Jitter: Audio only



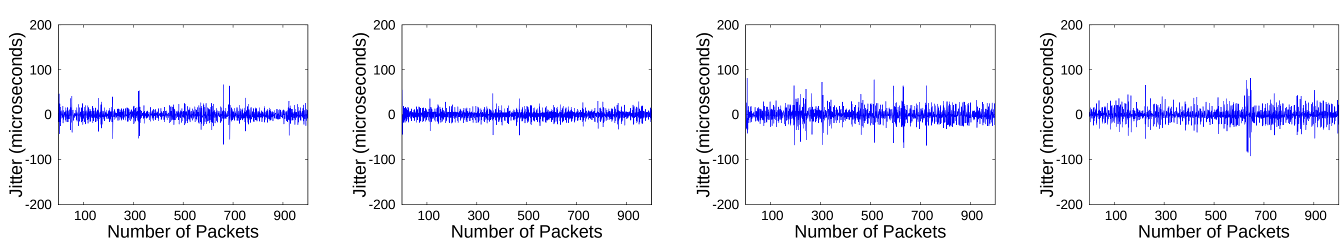
(a) Delay without CBS (b) Delay with CBS (c) Jitter without CBS (d) Jitter with CBS

Fig. 4: Cumulative Distribution Function of Per Packet Delay and Per Packet Jitter: Audio only



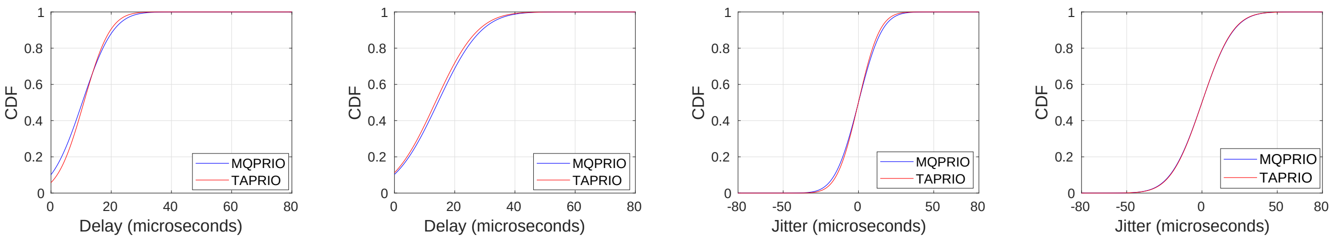
(a) MQPRIO: mix traffic (b) TAPRIO: mix traffic (c) CBS+MQPRIO: mix traffic (d) CBS+TAPRIO: mix traffic

Fig. 5: Per Packet Delay: Mix Traffic



(a) MQPRIO: mix traffic (b) TAPRIO: mix traffic (c) CBS+MQPRIO: mix traffic (d) CBS+TAPRIO: mix traffic

Fig. 6: Per Packet Jitter: Mix Traffic



(a) Delay without CBS (b) Delay with CBS (c) Jitter without CBS (d) Jitter with CBS

Fig. 7: Cumulative Distribution Function of Per Packet Delay and Per Packet Jitter: Mix Traffic

TAPRIO are the two queue disciplines that can map traffic flows to hardware queues, so we used CBS with TAPRIO and MQPRIO to perform the experiments. CBS with MQPRIO has an average delay of 20.785 microseconds, whereas CBS with TAPRIO has an average delay of 26.152 microseconds. TAPRIO works better for the time triggered and periodic traffic, and here we have used aperiodic audio traffic. CBS is a bandwidth reservation based algorithm. It allocates the traffic flow the bandwidth and transmits the flow according to the credits. So using MQPRIO with CBS is an excellent choice for directly assigning the priority, with MQPRIO handling the transmission. On the other hand, for CBS with TAPRIO, assigning the flow priority based on bandwidth, and then doing time based scheduling turns out to be an overhead for delay.

Figure 4 shows the Cumulative Distribution Function (CDF) of per packet delay and per packet jitter obtained for all the qdiscs for audio-only traffic. Figure 4b confirms that the delay with TAPRIO increases when CBS is used. Figures 4c and 4d show that jitter is 0 (i.e., there is no variation in the delay) for 50% of the times for all the qdiscs. This implies that all the qdiscs being evaluated in this work, with and without CBS, are effective in terms of delivering a deterministic behavior when the traffic is audio-only.

B. Mix traffic: Audio traffic with Best effort traffic

Figures 5a-5d and Figures 6a-6d depict the results obtained for per packet delay and jitter, respectively, for all the qdiscs.

The average per packet delay for MQPRIO is 15.125 microseconds and for TAPRIO is 10.706 microseconds, which follows a similar pattern when the traffic was audio-only. The per packet jitter of both TAPRIO and MQPRIO is in range of -100 to 100 microseconds, which shows bounded and low jitter characteristics of TSN traffic shaping algorithms. However, it can be seen that TAPRIO has more oscillations than MQPRIO because it employs time-triggered scheduling.

CBS with MQPRIO has an average delay of 12.975 microseconds, whereas CBS with TAPRIO has an average delay of 14.032 microseconds, which also follows the pattern that was observed with audio-only traffic. The performance of CBS with MQPRIO and TAPRIO in terms if jitter remains bounded within -100 to 100 microseconds, with the same observation that oscillations are less with TAPRIO.

Figure 7 shows the Cumulative Distribution Function (CDF) of per packet delay and per packet jitter obtained for all qdiscs with mix traffic. Figure 7b shows that when mix traffic is used, the delay with TAPRIO increases when CBS is used, but not to the extent as was the case when the traffic was audio-only. Figures 7c and 7d are similar to the ones shown in case of audio-only traffic; it can be observed that jitter is 0 (i.e., there is no variation in the delay) for 50% of the times for all the qdiscs. In fact, when it is a mix of audio traffic and best effort traffic, we notice that the performance of all the qdiscs is very similar to each other.

A noteworthy observation in these results is that the overall average numbers discussed above show a reverse trend when the nature of traffic shifts from audio-only to mix traffic. This is also apparent from the plots shown for both: audio-only traffic and mix traffic. One might expect that the average values would go up when the best traffic is introduced in the network. However, when the traffic is audio-only, the time-sensitive traffic competes with itself because both have equal priority, whereas when there is mix traffic, audio-traffic gets a higher priority over the best effort traffic.

V. Conclusions and Future Work

In this paper, we compared the performance of two prominent traffic shaping algorithms developed for Time Sensitive Networking. From the empirical evaluation, we can conclude that the delay and jitter values are bounded in all the experiments. Comparing the performance of MQPRIO and TAPRIO without using CBS, TAPRIO has lower average delays in both types of traffic (audio-only and mix). It also has lower oscillations in jitter than MQPRIO. When CBS is used with MQPRIO and TAPRIO, CBS with MQPRIO gives better results for both the performance parameters: per packet delays and jitter. A detailed evaluation of these algorithms on larger and more realistic topologies is a subject for future study.

Acknowledgments

The authors would like to thank Kaustubh Sharma, Nitin Dabi, Dayma Khan, Pallavi Gupta and Abhinaba Rakshit from the National Institute of Technology Karnataka, Surathkal, India for helping us with the flow of images used in this paper.

References

[1] IEEE-Standards Association et al. 802.1 Qav-2009-IEEE Standard for Local and Metropolitan Area Networks–Virtual Bridged Local Area Networks Amendment 12: Forwarding and Queuing Enhancements for Time-Sensitive Streams. Technical report, IEEE, 2009.

[2] IEEE-Standards Association et al. IEEE 802.1Qbv- Enhancements to Traffic Scheduling: Time-Aware Shaper (TAS). Technical report, IEEE, 2015.

[3] Lucia Lo Bello. Novel Trends in Automotive Networks: A Perspective on Ethernet and the IEEE Audio Video Bridging. *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, 2014.

[4] Jingyue Cao, Pieter JL Cuijpers, Reinder J Bril, and Johan J Lukkien. Tight Worst-case Response-time Analysis for Ethernet AVB using Eligible Intervals. In *2016 IEEE World Conference on Factory Communication Systems (WFCS)*, pages 1–8. IEEE, 2016.

[5] Jonas Diemer, Jonas Rox, and Rolf Ernst. Modeling of Ethernet AVB Networks for Worst-case Timing Analysis. *IFAC Proceedings Volumes*, 45(2):848–853, 2012.

[6] Frank Dürr and Naresh Ganesh Nayak. No-wait Packet Scheduling for IEEE Time-Sensitive Networks (TSN). In *Proceedings of the 24th International Conference on Real-Time Networks and Systems*, pages 203–212, 2016.

[7] John C Eidson, Mike Fischer, and Joe White. IEEE-1588™ Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Cystems. In *Proceedings of the 34th Annual Precise Time and Time Interval Systems and Applications Meeting*, pages 243–254, 2002.

[8] Norman Finn. Introduction to Time-Sensitive Networking. *IEEE Communications Standards Magazine*, 2(2):22–28, 2018.

[9] Xiaoyu Liu, Chi Xu, and Haibin Yu. Network Calculus-based Modeling of Time Sensitive Networking Shapers for Industrial Automation Networks. In *2019 11th International Conference on Wireless Communications and Signal Processing (WCSP)*, pages 1–7. IEEE, 2019.

[10] Philipp Meyer, Till Steinbach, Franz Korf, and Thomas C Schmidt. Extending IEEE 802.1 AVB with Time-triggered Scheduling: A Simulation Study of the Coexistence of Synchronous and Asynchronous Traffic. In *2013 IEEE Vehicular Networking Conference*, pages 47–54. IEEE, 2013.

[11] Ahmed Nasrallah, Akhilesh S Thyagaturu, Ziyad Alharbi, Cuixiang Wang, Xing Shao, Martin Reisslein, and Hesham Elbakoury. Performance Comparison of IEEE 802.1 TSN Time Aware Shaper (TAS) and Asynchronous Traffic Shaper (ATS). *IEEE Access*, 7, 2019.

[12] Ramon Serna Oliver, Silviu S Craciunas, and Wilfried Steiner. IEEE 802.1 Qbv Gate Control List Synthesis using Array Theory Encoding. In *2018 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 13–24. IEEE, 2018.

[13] Kurian Polachan, Chandramani Singh, and TV Prabhakar. Decentralized Dynamic Gate Scheduling of IEEE 802.1 Qbv Time Aware Shaper and a TSN simulator for Tactile Cyber-Physical Systems. In *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 45–53. IEEE, 2021.

[14] András Varga and Rudolf Hornig. An Overview of the OMNeT++ Simulation Environment. In *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, pages 1–10, 2008.

[15] Luxi Zhao, Paul Pop, and Sebastian Steinhorst. Quantitative Performance Comparison of Various Traffic Shapers in Time-Sensitive Networking. *arXiv preprint arXiv:2103.13424*, 2021.

[16] Luxi Zhao, Paul Pop, Zhong Zheng, Hugo Daigmorte, and Marc Boyer. Latency Analysis of Multiple Classes of AVB Traffic in TSN with Standard Credit Behavior using Network Calculus. *IEEE Transactions on Industrial Electronics*, 68(10):10291–10302, 2020.

[17] Luxi Zhao, Paul Pop, Zhong Zheng, and Qiao Li. Timing Analysis of AVB Traffic in TSN Networks using Network Calculus. In *2018 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 25–36. IEEE, 2018.