Simulative Evaluation of the TSN Mechanisms Time-Aware Shaper and Frame Preemption and Their Suitability for Industrial Use Cases

1st Anna Arestova
University of Erlangen-Nürnberg
Erlangen, Germany
anna.arestova@fau.de

2nd Kai-Steffen Jens Hielscher
University of Erlangen-Nürnberg
Erlangen, Germany
kai-steffen.hielscher@fau.de

3rd Reinhard German
University of Erlangen-Nürnberg
Erlangen, Germany
reinhard.german@fau.de

***Abstract*—Proprietary field buses such as PROFINET IRT, EtherNet/IP, and EtherCAT are currently used to ensure real-time communication in industrial automation appliances. Despite the fact that they are mostly Ethernet-based, interoperability between them is limited or not feasible at all. In the Industry 4.0 context, the goal is to enable real-time communication with ultra-low latencies, while also permitting non-critical traffic and ensuring interoperability between devices from different manufacturers. Time-Sensitive Networking (TSN) is a promising candidate to meet these requirements on layer 2 of the OSI reference model in conjunction with a suitable higher-level protocol. The IEEE 802.1 TSN Task Group has been developing extensions to the Ethernet technology to make it more deterministic and reliable since 2012. In this paper, we have evaluated the TSN mechanisms (1) Time-Aware Shaper (TAS) and (2) frame preemption (FP) and compared them to the existing strict priority scheduling (ST) mechanism in exemplary setups. The chosen comparison criteria are latency guarantee, jitter, and the influence on non-critical traffic. Since the support of TSN devices on the market is low, we have carried out the evaluation using the simulation framework OMNeT++ and the NeSTiNg library. Our evaluation shows that, all in all, TAS performs the best, but it introduces a great configuration overhead. FP and ST can keep up in certain scenarios and their performance can even be improved.**

***Index Terms*—TSN, industrial automation, real-time, TAS, frame preemption, strict priority**

I. Introduction

The topic of industrial communication is of particular relevance in today's world. Currently, well-known but typically slowly implemented efforts in automation technology, such as the convergence to Ethernet-based network technologies, are coming up against very far-reaching megatrends from the industry. Here, digitization, Industry 4.0, the use of cloud and edge computing, as well as, above all, flexible manufacturing are placing new and urgent demands on industrial communications in the near future. With regard to the technology that is currently in use, this results in the need for cross-manufacturer standardization, interoperability, and compatibility. Additionally, the coexistence of mixed-critical traffic should be allowed. Time-Sensitive Networking (TSN) [1] is particularly suitable for the requirements described because on the one hand it is built on the wide-spread and standardized Ethernet technology. On the other hand, TSN also provides deterministic and reliable behavior on the data link layer of the Open Systems Interconnection (OSI) reference model. The central and cooperative development within the IEEE 802.1 TSN Task Group promises the desired interoperability and gives hope for a comprehensive and common use in the future.

Strict timing requirements are especially encountered on the field level of industrial automation. Currently, several proprietary field buses like PROFINET and EtherCAT are represented on this level. These field buses particularly focus on guaranteeing deterministic and real-time behavior for time-sensitive traffic. Even though they are based on the Ethernet technology, they are not designed to interoperate with field buses from other vendors. Especially in the context of Industry 4.0 and Internet of Things (IoT), a high number of vendor-cross devices is expected to communicate in a best-effort and real-time pattern across all levels of the automation pyramid and up into the cloud [1], [2]. With TSN a common ground for devices of different manufacturers and network traffic with different criticality can be established.

TSN specifies latency control mechanisms such as Time-Aware Shaper (TAS) in IEEE 802.1Qbv [3] and frame preemption (FP) in IEEE 802.1Qbu [4]. These standards serve to limit latency bounds for dedicated traffic classes. While the former relies on precise scheduling approaches, the latter is based on a frame preemption principle. Since TSN is a modular system, plant operators are free to decide which TSN features to use. However, the decision depends on the requirements of the industrial communication network, like real-time, throughput of certain traffic classes, and efficiency of the communication technology. Therefore, we carry out an evaluation of TAS and FP regarding maximum end-to-end latency and jitter for time-critical traffic in this work. We additionally highlight the drawbacks, show optimization methods, and analyze the influence on the best-effort traffic. To put these features into context, we reproduce a test setup that can be found in industrial automation and compare our results to scenarios using strict priority scheduling (ST) that is applied in most Ethernet networks.

[1] https://1.ieee802.org/tsn/

ISBN 978-3-903176-39-3

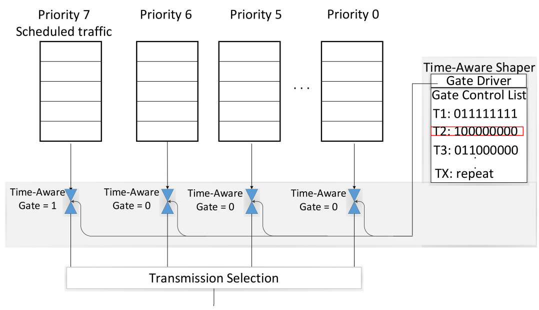


Fig. 1: TAS mechanism and components

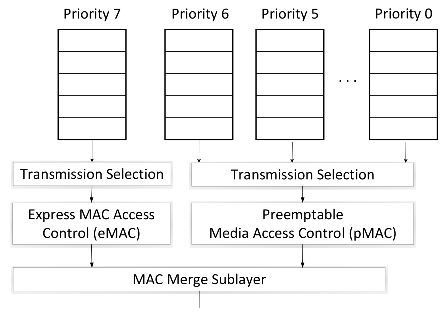


Fig. 2: Frame preemption MAC layers

II. Related Work

TSN is a popular research topic, especially in industrial automation. Many works regard TSN as a future technology for industrial communication [1], [2], [5]–[9]. TSN is able to interconnect heterogeneous devices [1], [2] and to deliver ultra-low latency and jitter [7], [10]. The most researched TSN standard is IEEE 802.1Qbv. The authors in [7], [10] demonstrate with small setups that the time-triggered transmission approach introduced in 802.1Qbv is capable to achieve ultra-low latency and jitter for time-sensitive traffic with submillisecond transmission cycles. We have simulated larger networks equipped with 802.1Qbv exchanging a higher number of time-critical streams compared to the mentioned publications. Our results also reveal that transmission cycles lower than 1 ms are achievable with TAS, even in a stress scenario. [11]–[13] show that a high number of TSN streams are schedulable when using 802.1Qbv. [8] states that the machine-to-machine protocol OPC UA in combination with TSN and TAS outperforms currently used Ethernet-based field buses. However, the authors state that the adoption of Gbps capable physical layer is the biggest advantage of TSN. Our experiments support this assumption.

Also, the frame preemption mechanism specified in IEEE 802.1Qbu is researched in the literature. [14]–[18] show that FP can compete with TAS regarding low latency and jitter guarantees. Our evaluation supports the thesis that FP can keep up with the performance of TAS, especially in scenarios with 1 Gbps links. But, as also stated in [17], [18], we have experienced that the performance of FP depends on the network size, link speed, and the amount of data sharing the same egress queue. In our 100 Mbps scenario, FP introduces a higher variation of jitter and thus leads to deadline misses for transmission cycles below 1 ms. However, we also show that the timing behavior of FP can be improved by defining proper transmission offsets to reduce the interference in egress queues. The author in [18] has carried out an extensive timing performance analysis for TAS and FP, primarily for smaller topologies and only a few time-critical senders. Our simulation setup is more extensive in number of network devices, time-critical senders, cycle times, and best-effort traffic generators. Hence, we are able to generate a high amount of cross-traffic of different and same criticality. [18] additionally shows that the use of cut-through instead of store-and-forward is another key aspect to reduce latency and jitter. We have only considered the store-and-forward because cut-through was not supported at the time of the evaluation.

Another beneficial setup for future communication systems is the utilization of FP in TAS guard bands, as addressed in [14], [19]. In our work, we also address and compare different guard band mechanisms to give a general idea about bandwidth usage in TAS setups.

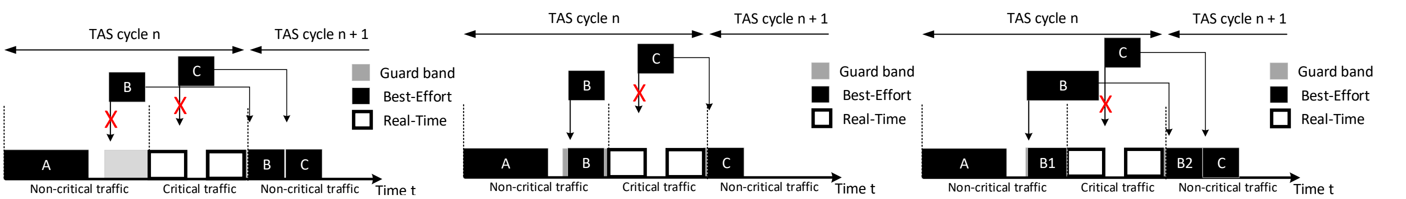
III. Fundamentals

A. Time-Aware Shaping

TAS is an emerging TSN mechanism that is specified in IEEE 802.1Qbv. Compared to other TSN standards, it is currently more in the focus of research and device manufacturing. TAS is located in egress ports of a switch or end-device interface. It relies on the classification of Ethernet packets into different traffic classes based on packet header fields. In Fig. 1, time-sensitive traffic is assigned the priority code point (PCP) value 7 in the Virtual Local Area Network (VLAN) tag and associated with a dedicated egress queue for scheduled traffic. The remaining PCP values can be specified for best-effort or e.g. audio and video traffic. TAS enables time-triggered injection of Ethernet frames on the communication medium. Therefore, the standard introduces time-aware *gates* for each egress queue that open and close according to a specified schedule. As shown in Fig. 1, at time T2 the queue for time-sensitive scheduled traffic is exclusively considered by the transmission selection as its gate is open. The slot ends at T3 and reopens in the next cycle. The gate schedule defined in the *Gate Control List* repeats after a defined cycle. If several gates are open at the same time, multiplexing according to the transmission selection algorithm, e.g. strict priority scheduling, is used among packets ready for transmission. One or several queues can be dedicated to time-critical traffic. We use exactly one queue for our use cases. By defining appropriate TAS schedules, time-sensitive streams can be guaranteed bounded latency and low jitter. However, the computation and configuration of all affected TAS gates in a communication network can result in a complex and time-consuming task. Furthermore, TAS requires a common notion of time for all involved devices.

TABLE I: Definitions and delay parameters

Parameter	Description	Dependency
$\mathrm{S_i}$	Stream $\mathrm{S_i} \in \mathrm{S}$ (set of critical streams)	
$\mathrm{N_j}$	Device node $\mathrm{N_j} \in \mathrm{N}$ (set of device nodes)	
$\mathrm{N_1}$	$\mathrm{N_1}$ is the talker node, the first device on a stream path	
$\mathrm{N_n}$	$\mathrm{N_n}$ is the listener node, the last device on a stream path	
$\mathrm{D^{trans}_{N_j,S_i}}$	Transmission delay of $\mathrm{S_i}$ on $\mathrm{N_j}$	Port speed, size of stream
$\mathrm{D^{trans}_{N_j,P_{BE}}}$	Remaining transmission delay of a best-effort packet $\mathrm{P_{BE}}$ on $\mathrm{N_j}$	Port speed, size of non-preemptable best-effort packet or fragment
$\mathrm{D^{queue}_{N_j,S_i}}$	Queuing delay of $\mathrm{S_i}$ on device $\mathrm{N_j}$	Port speed, number and size of predecessor streams in the same egress queue
$\mathrm{D^{proc}_{N_j}}$	Processing delay on switch device $\mathrm{N_j}$	Switch factory
$\mathrm{D^{prop}_{N_j,N_k}}$	Propagation delay on a transmission link between node $\mathrm{N_j}$ and $\mathrm{N_k}$	Length and material of cable
$\mathrm{D^{offset}_{S_i}}$	Start offset delay of $\mathrm{S_i}$ to prevent interference with other critical streams	Individual or automated scheduling
$\mathrm{D^{e2e}_{S_i}}$	End-to-end delay for $\mathrm{S_i}$	Aforementioned delays



(a) Fixed sized guard band (b) Flexible sized guard band (c) Guard band combined with FP

Fig. 3: Guard band mechanisms

Another important mechanism introduced in IEEE 802.1Qbv is the *guard band*. Each queue is assigned a guard band that prevents the transmission of packets for a fixed configured amount of time (Fig. 3a) or a flexible amount of time (Fig. 3b) before launching a critical slot. The latter permits the transmission if the frame does not intrude into the subsequent scheduled slot. [4] introduces the combination of guard bands and FP. Thus, the size of the guard band can be reduced to the maximum fragment size of the FP mechanism: 127 octets. As Fig. 3c shows, one part of packet B can still be transmitted while this is not possible with other guard band mechanisms.

B. Frame Preemption

The combination of the standards IEEE 802.3br (specification and management parameters for interspersing express traffic) [20] and the TSN standard IEEE 802.1Qbu (frame preemption) [4] allows fragmentation of Ethernet packets into smaller framelets and their subsequent composition, as well as the preemption of certain traffic classes in favor of other traffic classes. IEEE 802.3br introduces two Media Access Layer (MAC) service interfaces: a preemptable MAC (pMAC) service interface and an express MAC (eMAC) service interface [4], see Fig. 2. We call the egress queues connected to eMAC *express queues* and the others *preemptable queues*. The additional service interfaces are located below the transmission selection. Thus, the transmission selection can operate e.g. according to the strict priority principle. The MAC Merge sublayer enables the preemption of non-express traffic being in transmission or before the start of the transmission [20]. However, frame preemption cannot happen at any point of the preemptable frame. The following rules apply [4]:

- The final fragment size must not be smaller than 64 octets without considering further delays[2].
- The worst-case delay of a ready-for-transmission express frame caused by a preemptable frame is smaller than 128 octets times.
- Frames from express queues cannot preempt another express frame.
- Frames from preemptable queues cannot preempt another preemptable frame.

IV. Formal Delay and Jitter Consideration

Before carrying out our simulation evaluations, we want to show how latency and jitter are caused when using different link layer mechanisms. When using TAS, it is possible to calculate precise schedules that minimize the jitter and latency of a time-sensitive stream. This is especially the case when the applied scheduling algorithm ensures that each stream is isolated in all egress queues on its path and the same stream order can be maintained over the whole runtime with the help of computed start offsets for the stream [11]–[13]. Traffic control mechanisms such as Earliest TxTime First (ETF) allow time-triggered handover of frames to network interface cards in end devices based on those those offsets. In case of stream isolation in each high-priority queue, the end-to-end latency of a Stream $\mathrm{S_i}$ is composed of the static network delays $\mathrm{D^{trans}_{N_J,S_i}}$ and $\mathrm{D^{proc}_{N_j}}$ (Tab. I) for each crossed network and end-device

[2]Overhead delays caused by preamble, start frame delimiter (SFD), and inter-frame gap (IFG) are excluded.

TABLE II: Description of simulation experiments

Experiment	Description	Transmission Rate	Special setups
TASStat	TAS and ST were configured in all TSN capable devices	1 Gbps/100 Mbps	Static-sized guard bands
TASFlex	TAS and ST were configured in all TSN capable devices	1 Gbps/100 Mbps	Variable-sized guard bands
TASFP	TAS and ST were configured in all TSN capable devices	1 Gbps/100 Mbps	Guard bands combined with FP
FP	FP combined with ST was used in all TSN capable devices	1 Gbps/100 Mbps	/
ST	Only ST was used in all devices	1 Gbps/100 Mbps	/
FPO	FP combined with ST was used in all TSN capable devices	1 Gbps/100 Mbps	Transmission offsets are optimized
STO	Only ST was used in all devices	1 Gbps/100 Mbps	Transmission offsets are optimized

$\mathrm{N_j}$, and $\mathrm{D^{prop}_{N_j,N_k}}$ for each passed transmission link between adjacent devices $\mathrm{N_j}$ and $\mathrm{N_k}$. An additional offset delay $\mathrm{D^{offset}_{S_i}}$ has to be considered. Thus, the end-to-end delay for a unicast stream $\mathrm{S_i}$ in a TAS scenario and store-and-forward switches results in Eq. 1.

$$\mathrm{D^{e2e}_{S_i}} = \mathrm{D^{offset}_{S_i}} + \mathrm{D^{trans}_{N_1,S_i}} + \mathrm{D^{prop}_{N_1,N_2}} + \sum_{j=2}^{n-1} \mathrm{D^{proc}_{N_j}} + \mathrm{D^{trans}_{N_j,S_i}} + \mathrm{D^{prop}_{N_j,N_{j+1}}} \tag{1}$$

When using FP combined with ST or only ST, time-sensitive streams may have to encounter inference delay caused by a lower-priority frame in transmission ($\mathrm{D^{trans}_{N_j,P_{BE}}}$) and queuing delay summoned by streams of the same priority ($\mathrm{D^{queue}_{N_j,S_i}}$) in each passed device. The end-to-end delay for stream $\mathrm{S_i}$ in FP combined with ST and only ST setups with store-and-forward switches is described in Eq. 2. This assumption applies if the time-sensitive traffic is assigned the highest-priority and no special transmission offsets are specified.

$$\begin{aligned}\mathrm{D^{e2e}_{S_i}} = {} & \mathrm{D^{queue}_{N_1,S_i}} + \mathrm{D^{trans}_{N_1,P_{BE}}} + \mathrm{D^{trans}_{N_1,S_i}} + \mathrm{D^{prop}_{N_1,N_2}} + \sum_{j=2}^{n-1} \mathrm{D^{proc}_{N_j}} \\ & + \mathrm{D^{trans}_{N_j,S_i}} + \mathrm{D^{prop}_{N_j,N_{i+1}}} + \mathrm{D^{queue}_{N_j,S_i}} + \mathrm{D^{trans}_{N_j,P_{BE}}}\end{aligned} \tag{2}$$

In ST setups, $\mathrm{D^{trans}_{N_j,P_{BE}}}$ can have the size of a maximum transmission unit that results in 1542 octets on wire. In scenarios using FP combined with ST, $\mathrm{D^{trans}_{N_j,P_{BE}}}$ can be assumed the size of a maximum non-preemptable unit that is about 12 times smaller than a maximum transmission unit. However, the queuing delay cannot be reduced in FP scenarios. $\mathrm{D^{trans}_{N_j,P_{BE}}}$ and $\mathrm{D^{queue}_{N_j,S_i}}$ are the main contributors for end-to-end latency variations. Consequently, a larger number of hops and a lower link speed increase the jitter. Defining proper send offsets can help reducing jitter in queues but it does not completely avoid collisions with other frames. Nonetheless, the expected results for maximum end-to-end latency when using FP and ST can be analyzed with theoretical frameworks like *network calculus*. In this way, the suitability of FP combined with ST and only ST can be verified for certain network topologies.

V. Evaluation

A. Test Setup and Configuration

We have carried out a simulative analysis of TAS, FP, and ST in OMNeT++[3] using the INET[4] and NeSTiNg[5] [21] library. The experiments are described in Tab. II. We have picked a network architecture that is typical for industrial automation using Ethernet technology. In our case, the network topology is a tree topology, i.e. hierarchical interconnection of several star topologies, see Fig. 4. The architecture shown in Fig. 4 mirrors an excerpt from the field and control level of factory automation where programmable logic controllers (PLCs) exchange time-critical control and status signals with other PLCs, with I/O devices, or directly with actuators and sensors. The actually simulated topology is shown in Fig. 5. All switches (rectangles) and named end-devices (circles) can be equipped with TAS, FP, and ST. Unnamed nodes only support basic Ethernet functions. All devices have a common notion of time and all switches support store-and-forward.

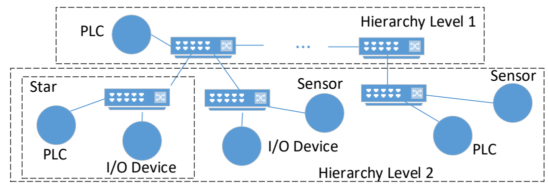


Fig. 4: Exemplary tree topology

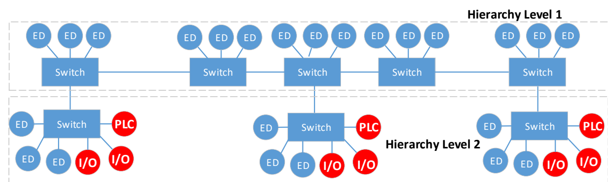
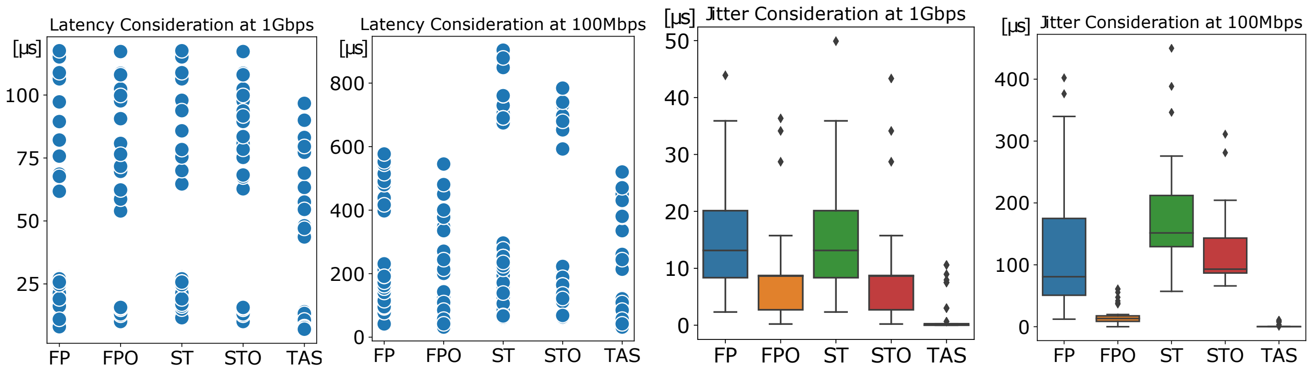


Fig. 5: Simulated topology in OMNeT++

Named nodes (PLC, I/O) exchange time-sensitive traffic with different transmission cycle times in the range of $\{0.5\,\mathrm{ms}, 2\,\mathrm{ms}, 4\,\mathrm{ms}\}$ and over 1, 5, or 7 switches. Each time-sensitive stream carries one Ethernet packet with the size of 100 octets for hierarchy level 2 and 200 to 500 octets for hierarchy level 1. 50 time-sensitive streams are equally distributed to named nodes and exchanged in a multicast or unicast fashion, 20 on level 1 and 30 on level 2. Best-effort streams broadcast to all devices in a bursty mode to create a worst-case scenario for critical streams. Time-sensitive traffic occupies one traffic queue and is given the PCP value 7. We have generated a network load of up to 70%. To evaluate the influence of the transmission rate of links, we have run separate tests with 1 Gbps and 100 Mbps Ethernet links. TAS depends on a proper

[3]https://www.omnetpp.org
[4]https://inet.omnetpp.org/Introduction.html
[5]https://gitlab.com/ipvs/nesting



(a) Latency 1 Gbps (b) Latency 100 Mbps (c) Jitter 1 Gbps (d) Jitter 100 Mbps

Fig. 6: Maximum end-to-end latency and jitter for 1 Gbps and 100 Mbps scenarios.

examination of TSN stream requirements and capabilities of network and end devices. One such prerequisite of a TSN stream is e.g. a bounded end-to-end network transmission latency, excluding application and operating system overhead, and low jitter. We begin the delay measurement when the packet arrives at the egress queue of the sending host. In our case, we require that the end-to-end latency is less than or equal to the transmission cycle time. We have applied the meta-heuristic from [13] to calculate a feasible schedule that determines the sending offset of critical streams within their transmission cycle and how the TASs of all affected devices have to be configured. For FP, we set the egress queues associated with priority 7 to express queues and the remaining to preemptable queues. We have defined two variants for FP and ST: an unoptimized (FP and ST) and an optimized (FPO and STO). In the unoptimized version, all time-sensitive streams are ready for transmission at the beginning of their transmission cycle. The optimized version adapts the start transmission offsets of TAS in order to investigate if stream isolation can reduce the collision of high-priority traffic in egress queues for FP and ST. To achieve good results, it is required that all end devices are time-synchronized. The transmission selection algorithm in TAS, FP, and FPO is ST.

B. Results for Latency and Jitter

We have measured the maximum end-to-end latency (Fig. 6a and Fig. 6b) and maximum jitter (Fig. 6c and Fig. 6d) for all real-time streams with different transmission cycles, frame sizes, and paths in several simulation runs. All deadlines were met in the 1 Gbps scenario. TAS (here TASStat) illustrates the lowest latency and jitter. FPO and STO perform slightly better than FP and ST. The experienced jitter for FP, FPO, ST, and STO is higher than for TAS, but also acceptable. In the 100 Mbps scenario, TAS and FPO fulfilled all end-to-end latency requirements whereas FP, ST, and STO missed a few deadlines for streams with $0.5\,\mathrm{ms}$ transmission cycle that are transmitted over the hierarchy level 1. FPO and STO outperform FP and ST in terms of latency and especially jitter for 100 Mbps scenarios. TAS maintains low jitter statistics even for 100 Mbps links.

TABLE III: Percentage deviation of throughput of best-effort traffic compared to ST setups.

Link Layer Mechanism	1 Gbps [%]	100 Mbps [%]
TASStat	$< 0.001\%$	$< 0.005\%$
TASFlex	$< 0.001\%$	$< 0.002\%$
TASFP	$< 0.001\%$	$< 0.001\%$
FP	$< 0.001\%$	$< 0.001\%$

Even if TAS shows the best performance, FP, FPO, STO, and ST prove to be suitable for 1 Gbps scenarios. The evaluation shows that FPO and STO improve the timing behavior of ST and FP, especially by reducing the jitter. A more suitable scheduling algorithm may have provided even better results. However, the simulation runs may not demonstrate the actual worst-case latency and jitter. Thus, a detailed analytical investigation has to be carried out for non-TAS layers. Among others, clock deviation was not considered in our simulation. This would contribute to greater jitter for all mechanisms. It is also important to annotate that TAS, FPO, and STO provide configuration overhead and depend on time-synchronization.

C. Throughput and Bandwidth Efficiency

To evaluate how the link layer mechanisms impact non-critical traffic, we have tracked the throughput of best-effort packets in all simulation scenarios. Therefore, we have generated a huge amount of best-effort traffic and analyzed the arriving best-effort traffic in all unnamed hosts in bit/sec. The highest throughput was recorded in ST scenarios. Our results in Tab. III shows the deviation of throughput compared to ST. In 1 Gbps scenarios, the throughput was similar. In 100 Mbps setups, the difference was a bit higher in case of TAS with fixed and flexible guard bands, but not significant. The impact of the guard band mechanism depends on the number of configured critical gates in a network and accordingly on the scheduling mechanism. In our case, the number of slots was low and optimized.

Moreover, we have taken a closer look at the guard band mechanisms introduced in Sec. III-A in the experiments TASStat, TASFlex, and TASFP. To create statistics, we have recorded the amount of best-effort traffic that was prevented from transmission due to guard bands. We have figured out that in TASStat the counter was about 4 times bigger than in TASFlex and about 18 times bigger than in TASFP. Thus, TASFP shows the best bandwidth efficiency but requires additional MAC control layers. However, the amount of prevented traffic was not significant. In addition to the guard bands, unused bandwidth can also be located in critical TAS slots if the schedule is not optimal.

VI. Conclusion

In this work, we have compared the TAS and FP mechnisms of TSN against the existing ST link layer scheduling approach in the simulation environment OMNeT++. We have considered the maximum end-to-end latency, maximum jitter, throughput of non-critical traffic, and bandwidth usage in TAS with exemplary test setups. The result is that all mechanisms show a good performance in 1 Gbps scenarios regarding maximum end-to-end latency and maximum jitter. TAS is particularly suitable for applications where ultra-low jitter and ultra-low latency are required, such as motion control, independent of transmission link speed. Unfortunately, a drawback of TAS is the configuration effort and the unused bandwidth caused by the guard band mechanism. Yet our evaluation showed that the influence of guard bands is not significant under certain conditions. Bandwidth wastage can also be reduced when combining guard band mechanism with FP. The FP mechanism promises bounded jitter and latency behavior for transmission cycles in the lower millisecond range or slightly below for 1 Gbps and 100 Mbps. Thus, FP is also a suitable candidate for real-time systems after a careful timing analysis, e.g. using analytical methods. Moreover, FP requires less configuration effort than TAS. Unfortunately, FP has limited availability on the market. ST also provides good results in the latency and jitter analysis in 1 Gbps setups. However, the jitter is high in 100 Mbps scenarios. ST can be used in real-time environments with lower milliseconds cycles, but requires careful considerations under worst-case network load. Nonetheless, ST shows the least configuration overhead, and devices supporting ST are highly available on the market. Finally, our evaluation has shown that the timing behavior of ST and FP can be improved by computing transmission offsets that help reducing the interference in egress queues. We want to examine this behavior analytically in future works.

References

[1] P. Drahoš, E. Kučera, O. Haffner, and I. Klimo, "Trends in industrial communication and OPC UA," in *2018 Cybernetics Informatics (K I)*, Jan. 2018, pp. 1–5.

[2] F. Zezulka, P. Marcon, Z. Bradac, J. Arm, T. Benesl, and I. Vesely, "Communication Systems for Industry 4.0 and the IIoT," *IFAC-PapersOnLine*, vol. 51, no. 6, pp. 150–155, 2018, 15th IFAC Conference on Programmable Devices and Embedded Systems PDeS 2018.

[3] IEEE Computer Society, "IEEE Standard for Local and Metropolitan Area Networks – Bridges and Bridged Networks – Amendment 25: Enhancements for Scheduled Traffic," *IEEE Std 802.1Qbv-2015 (Amendment to IEEE Std 802.1Q-2014 as amended by IEEE Std 802.1Qca-2015, IEEE Std 802.1Qcd-2015, and IEEE Std 802.1Q-2014/Cor 1-2015)*, Mar. 2016.

[4] ——, "IEEE Standard for Local and Metropolitan Area Networks – Bridges and Bridged Networks – Amendment 26: Frame Preemption," *IEEE Std 802.1Qbu-2016 (Amendment to IEEE Std 802.1Q-2014)*, Aug. 2016.

[5] D. Bruckner, M. Stănică, R. Blair, S. Schriegel, S. Kehrer, M. Seewald, and T. Sauter, "An Introduction to OPC UA TSN for Industrial Communication Systems," *Proceedings of the IEEE*, vol. 107, no. 6, pp. 1121–1131, 2019.

[6] S. K. Panda, M. Majumder, L. Wisniewski, and J. Jasperneite, "Real-time Industrial Communication by using OPC UA Field Level Communication," in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, 2020, pp. 1143–1146.

[7] J. Pfrommer, A. Ebner, S. Ravikumar, and B. Karunakaran, "Open source OPC UA PubSub over TSN for Realtime Industrial Communication," in *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 2018, pp. 1087–1090.

[8] D. Bruckner, R. Blair, M.-P. Stanica *et al.*, "OPC UA TSN. A new Solution for Industrial Communication," 2018.

[9] P. Pop, M. L. Raagaard, M. Gutierrez, and W. Steiner, "Enabling Fog Computing for Industrial Automation Through Time-Sensitive Networking (TSN)," *IEEE Communications Standards Magazine*, vol. 2, no. 2, pp. 55–61, 2018.

[10] M. H. Farzaneh and A. Knoll, "Time-sensitive networking (tsn): An experimental setup," in *2017 IEEE Vehicular Networking Conference (VNC)*, 2017, pp. 23–26.

[11] F. Dürr and N. G. Nayak, "No-wait Packet Scheduling for IEEE Time-sensitive Networks (TSN)," in *Proceedings of the 24th International Conference on Real-Time Networks and Systems*, ser. RTNS '16. New York, NY, USA: ACM, 2016, pp. 203–212.

[12] S. S. Craciunas, R. S. Oliver, M. Chmelík, and W. Steiner, "Scheduling Real-Time Communication in IEEE 802.1Qbv Time Sensitive Networks," in *Proceedings of the 24th International Conference on Real-Time Networks and Systems*, ser. RTNS '16. New York, NY, USA: ACM, 2016, pp. 183–192.

[13] A. Arestova, K.-S. J. Hielscher, and R. German, "Design of a Hybrid Genetic Algorithm for Time-Sensitive Networking," in *Measurement, Modelling and Evaluation of Computing Systems*. Springer International Publishing, Mar. 2020, pp. 99–117.

[14] D. Thiele and R. Ernst, "Formal worst-case performance analysis of time-sensitive ethernet with frame preemption," in *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2016, pp. 1–9.

[15] A. Gogolev and P. Bauer, "A simpler TSN? Traffic Scheduling vs. Preemption." in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, 2020, pp. 183–189.

[16] Zifan Zhou, Ying Yan, S. Ruepp, and M. Berger, "Analysis and implementation of packet preemption for time sensitive networks," in *2017 IEEE 18th International Conference on High Performance Switching and Routing (HPSR)*, 2017, pp. 1–6.

[17] D. Hellmanns, J. Falk, A. Glavackij, R. Hummen, S. Kehrer, and F. Dürr, "On the performance of stream-based, class-based time-aware shaping and frame preemption in tsn," in *2020 IEEE International Conference on Industrial Technology (ICIT)*, 2020, pp. 298–303.

[18] S. Nsaibi, "Timing Performance Analysis of the Deterministic Ethernet Enhancements Time-Sensitive Networking (TSN) for Use in the Industrial Communication," doctoralthesis, Technische Universität Kaiserslautern, 2020. [Online]. Available: http://nbn-resolving.de/urn:nbn:de:hbz:386-kluedo-59955

[19] M. Kovacevic, V. Skobic, M. Knezic, and Z. Ivanovic, "Towards Implementation of Frame Preemption Mechanism on FPGA Platform," in *2020 19th International Symposium INFOTEH-JAHORINA (INFOTEH)*, 2020, pp. 1–7.

[20] IEEE Computer Society, "IEEE Standard for Ethernet Amendment 5: Specification and Management Parameters for Interspersing Express Traffic," *IEEE Std 802.3br-2016 (Amendment to IEEE Std 802.3-2015 as amended by IEEE St802.3bw-2015, IEEE Std 802.3by-2016, IEEE Std 802.3bq-2016, and IEEE Std 802.3bp-2016)*, Oct. 2016.

[21] J. Falk, D. Hellmanns, B. Carabelli, N. Nayak, F. Dürr, S. Kehrer, and K. Rothermel, "NeSTiNg: Simulating IEEE Time-sensitive Networking (TSN) in OMNeT++," in *2019 International Conference on Networked Systems (NetSys)*, 2019, pp. 1–8.