

Impact of Packet Loss and Timing Errors on Scheduled Periodic Traffic with Time-Aware Shaping (TAS) in Time-Sensitive Networking (TSN)

Manuel Eppler, Steffen Lindner, Lukas Osswald, Thomas Stüber, Michael Menth

Chair of Communication Networks, University of Tuebingen, Germany

{manuel.eppler, steffen.lindner, lukas.osswald, thomas.stueber, menth}@uni-tuebingen.de

Abstract—Time-Sensitive Networking (TSN) is a collection of mechanisms to enhance the realtime transmission capability of Ethernet networks. TSN combines priority queuing, traffic scheduling, and the Time-Aware Shaper (TAS) to carry periodic traffic with ultra-low latency and jitter. That is, so-called Talkers send periodic traffic with highest priority according to a schedule. The schedule is designed such that the scheduled traffic is forwarded by the TSN bridges with no or only little queuing delay. To protect that traffic against other frames, the TAS is configured on all interfaces such that lower-priority queues can send only when high-priority traffic is not supposed to be forwarded. In the literature on scheduling algorithms for the TAS there is mostly the explicit or implicit assumption that the TAS also limits transmission slots of high-priority traffic.

In this paper we show that this assumption can lead to tremendous problems like very long queuing delay or even packet loss in case of faulty frames. A faulty frame arrives too early or too late according to the schedule, it is missing or additional. We construct minimal examples to illustrate basic effects of faulty frames on a single link and demonstrate how this effect can propagate through the networks and cause remote problems. We further show using simulations that a single slightly delayed frame may lead to frame loss on multiple links. We show that these problems can be alleviated or avoided when TAS-based transmission slots for high-priority traffic are configured longer than needed or if they are not limited at all.

Index Terms—TSN, Reliability, TAS, PSFP, OMNET++, Scheduled Traffic

A. List of frequently used acronyms

We use the following acronyms.

ATS	Asynchronous Traffic Shaping
BE	Best Effort
CBS	Credit-Based Shaper
CNC	Centralized Network Configuration
GCL	Gate Control List
gPTP	generalized Precision Time Protocol
IPV	Internal Priority Value
QoS	Quality of Service
TAS	Time-Aware Shaper
TSN	Time-Sensitive Networking
PSST	Per-Stream Scheduled Traffic
PCP	Priority Code Point
PSFP	Per-Stream Filtering and Policing
EST	Enhancements for Scheduled Traffic
TDMA	Time Division Multiple Access
TSA	Transmission Selection Algorithm

I. INTRODUCTION

The rise of Industry 4.0 applications, e.g., intelligent factory automation, results in steadily increasing demands on communication quality. Real-time applications, e.g., controlling the movement of a robotic arm, require guaranteed sub-microsecond end-to-end delays across multiple network nodes, so-called isochronous traffic. As conventional Ethernet technology cannot meet these stringent Quality of Service (QoS) requirements, the Time-Sensitive Networking (TSN) standards has been evolved to provide ultra-low latency, zero jitter, and no congestion-based packet loss. TSN consists of standards for time synchronization, deterministic latencies, and high reliability. TSN defines generalized Precision Time Protocol (gPTP), a configuration for the packet based synchronization protocol PTP that allows a nanosecond precise time synchronization of all TSN devices. This is the basis for any synchronous operation. The enhancements for scheduled traffic IEEE 802.1Qbv [1] enable the Time-Aware Shaper (TAS) for deterministic forwarding with ultra-low delay. It consists of eight transmission gates that control the transmission selection from the eight priority queues in the egress port. While a gate is closed, the queue is prevented from sending, until the gate opens. The so-called Gate Control List (GCL) defines the timing-based state of the gates. By closing all gates but one, the TAS can provide exclusive link access for a single priority omitting any delay from an already started transmission of a lower priority packet. By carefully planning each transmission time at the Talkers and forwarding delay at the switches, TSN implements a per-stream scheduled traffic that omits any congestion based delay. The calculation of this TAS schedule in combination with the precise sending time of each stream at the end stations is based on different information, such as the QoS requirements, network topology, and traffic description. The Industrial Internet Consortium (IIC) published the traffic types of Industrial Automation and Control Systems and maps them to the TSN technologies. The IIC classifies critical applications into 3 classes: Isochronous, Cyclic, and Event-based traffic. This paper focuses on the Isochronous traffic. The calculated schedule depends on the correct arrival time of frames at bridges. Missing, additional, early, or late frames might lead to too long end-to-end delays or packet loss, either for the end stations that deviated from their planned sending behavior or even for other end stations that strictly

adhered to their planned sending behavior. The effects of misconfigured or malfunctioning end stations for network-wide TAS schedules have not yet been investigated. The lack of knowledge about the fragility of Per-Stream Scheduled Traffic in TSN can lead to essential security measures being removed as a cost-cutting measure, resulting in fatal system failures. The paper states four main contributions. First, it defines the general problem statement of Per-Stream Scheduled Traffic (PSST) with misconfigured or malfunctioning end stations. Therefore, it presents a visualization methodology for TAS schedules over time. It explains the effects of faulty frames, i.e., frames that are missing, additional, early, or late, and visualizes the consequences for the TAS schedule. Second, it quantifies the probability of errors based on faulty frames for network-wide TAS schedules via simulation. Finally, it demonstrates and simulates solution approaches that illustrate the necessity of time-based Per-Stream Filtering and Policing (PSFP) to resolve the fragility issue of TAS.

The paper is structured as follows. We review related work in Section II and introduce background information on TSN in Section III. Section IV explains the problem statement and presents our novel visualization methodology for TAS schedules. We explain and visualize the effects of faulty frames in Section V and present preliminary simulation results in Section VI. Finally, we conclude the paper in Section VII.

II. RELATED WORK

The TSN standard "IEEE Std. 802.1Qbv Enhancements for scheduled traffic" was first published in 2018. In the last 6 years, a lot of researchers published algorithms to create TSN schedules [2]. Some of them implement reliability features within their algorithms based on specific fault models [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17]. Those reliability features are designed for specific faults and can be structured according to their countermeasures. Dobirn et al. [10] as well as Park et al. [11] build their schedule to enable re-transmissions of missing frames within the frames deadline. Therefore, they schedule enough time between frames, such that re-transmissions do not interfere with other streams. A similar approach that copes with frame loss is by Feng et al. [5], [9] and Reusch et al. [7]. They send redundant copies of frames over the same path to ensure that the stream data is received even if a single frame is missing. Zhou et al. [13] step further by adding a routing mechanism, such that the copies are no longer routed over the same path based on the probability of a frame loss, which does not necessarily result in disjoint paths. In contrast, Atallah et al. [6], [12], Gavrilut et al. [8], Syed et al. [17], [18], and Huang et al. [3] ensure redundant copies that are sent via disjoint paths. Only two of those used the IEEE Std. 802.1CB Frame Replication and Elimination for Reliability [19] for this approach [3], [17]. Syed et al. [20] present an encoding scheme to reduce the network capacity needed for redundancy information. They use the XOR of two streams as a redundancy stream, which is sent via a path, disjoint of both other streams. Therefore if one link fails, the information from the remaining two streams can be used to reconstruct

the missing stream. Another approach is fast rescheduling and rerouting after a link failure, which is implemented by Pozo et al. [21].

Crae et al. defined the Frame and Flow Isolation [22], which is implemented by some scheduling algorithms [23] (Frame Isolation [13]).

Craciunas et al. introduce a scheduling that is robust against clock drifts [4]. Therefore, small sending time variations would not lead to interference between frames.

Zhou et al. implement a hardware selection that selects the TSN hardware based on its costs, tests, and the reliability requirements of the streams [15].

5G ACIA white paper lists typical use cases for TSN in factory environments as well as requirements of industrial communication. The white paper states two kinds of scheduled traffic called Isochronous and Cyclic-Synchronous. For both scheduled traffic types, IEEE802.1 Qbv and Qci are mentioned as mandatory without any reasoning. Although Qci is mentioned as mandatory, according to the IEC/IEEE 60802, it is currently an optional feature for TSN Bridges.

III. TIME-SENSITIVE NETWORKING (TSN)

We first give an overview of Time-Sensitive Networking (TSN) and explain the TAS. Afterwards we focus on Per-Stream Scheduled Traffic (PSST) and the PSFP.

A. Overview

Time-Sensitive Networking is an enhancement of Ethernet and is currently under standardization by the IEEE 802.1 TSN task group. TSN specifies shaping and policing mechanisms that enable data transmission with QoS requirements. Examples for QoS requirements are bounded jitter, zero congestion-based packet loss, and sub-microsecond end-to-end delays. End stations in a TSN network that transmit data are called talkers, and end stations that receive data are called listeners. Talkers and listeners are connected through possibly multiple TSN bridges that apply shaping and policing mechanisms to provide QoS. Examples for shaping and policing mechanisms are Credit-Based Shaper (CBS) [24], Per-Stream Filtering and Policing (PSFP) [25], Time-Aware Shaper (TAS) [1], and Asynchronous Traffic Shaping (ATS) [26].

End stations signal their QoS requirements, e.g., maximum acceptable latency, to the network before they start data transmission. Further, they describe their data transmission in the form of an interval, also called a period, the maximum number of frames that are transmitted during the interval, and the maximum frame size. This allows the calculation of an upper bound on the required datarate. A logical data transmission between a talker and (multiple) listeners is called a TSN stream. TSN streams are admitted to the network if the QoS demands of both the talker and listeners can be met. This admission control decision is either done in a distributed manner or by a central configuration entity, the so-called Centralized Network Configuration (CNC). Finally, bridges on the path between the talker and listeners are configured such that the previously signaled QoS demands are fulfilled.

B. Time-Aware Shaper (TAS)

The policing and shaping mechanisms can provide different levels of QoS guarantees. For example, the Credit-Based Shaper (CBS) provides a guaranteed bandwidth share but cannot guarantee sub-microsecond end-to-end delays. IEEE 802.1Qbv [1] specifies the enhancements for schedule traffic, often referred to as the Time-Aware Shaper (TAS). TAS enables sub-microsecond end-to-end delays for time-critical real-time communication in TSN networks. It is based on a gating mechanism that introduces time slots in which frames are eligible for transmission, similar to Time Division Multiple Access (TDMA) [27]. Figure 1 illustrates the structure of the Queuing mechanism in a TSN compliant bridge with TAS.

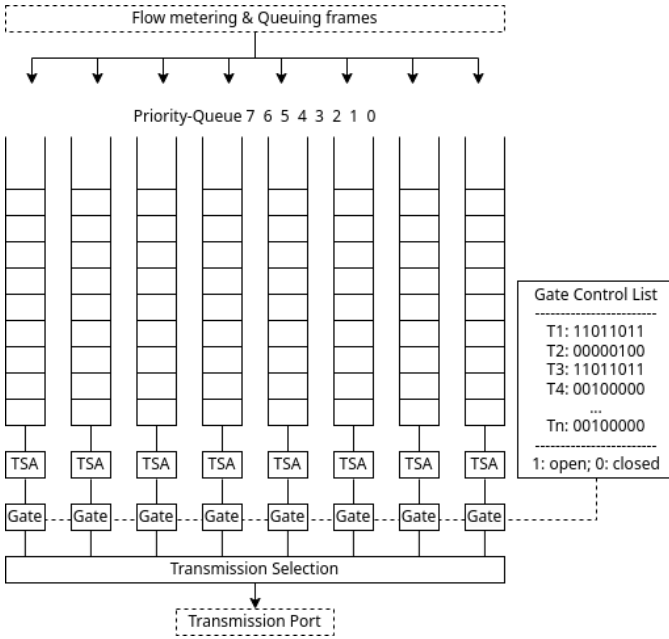


Fig. 1. Priority queuing and gate mechanism of the IEEE802.1Qbv Enhancements for scheduled traffic.

TSN bridges support up to eight priority queues per egress port. Frames of a TSN stream are enqueued in a specific queue based on the Priority Code Point (PCP) value of its VLAN tag. A priority queue is followed by a Transmission Selection Algorithm (TSA), e.g., CBS, and a transmission gate with two possible states: open and close. Frames are only eligible for transmission if their corresponding gate is open. The GCL controls a periodic state change of all gates, called schedule. A GCL entry maps a relative time interval $[T_i, T_{i+1}]$ to a bit vector, where each bit corresponds to a single transmission gate. If the bit is set, the gate is open. Otherwise, it is closed. The schedule repeats periodically for an indefinite number of times. The length of a single schedule is called the hyperperiod, which is typically the least common multiple of all stream periods. If multiple queues are eligible for transmission, i.e., several transmission gates are open, the queue with the highest priority is used.

Prioritizing or delaying streams can be done by closing the gates of other priorities at the right time. When multiple bridge

schedules are planned together, sub-microsecond end-to-end delays can be achieved.

C. Per-Stream Scheduled Traffic (PSST)

To support streams with sub-microsecond end-to-end delays, frames need to arrive at the bridge in time, i.e., when the corresponding transmission gate is open. Otherwise, the frame is received while the gate is closed. This introduces additional queuing delay, which may violate the stream's deadline. Consequently, both the end stations and the network require precise time synchronization, e.g., with gPTP IEEE Std. 802.1AS [28]. Additionally, the sending time at the talker and the deployed TAS schedule needs to be compatible. When the talker signals its QoS requirements, it additionally sends its earliest and latest possible sending time, called transmit offset (ETO and LTO), as well as its synchronization jitter to the CNC. Based on these information, the CNC calculates the sending offset for the talker and an appropriate GCL for all bridges on the path, such that the previously signaled QoS requirements are met. The resulting schedule may or may not include queuing at some bridges. Section IV illustrates the consequences of different schedule decisions.

Calculating such a schedule is NP-complete and can be done by suitable algorithms [2]. The combination of sending offsets at talkers and TAS within the network is called Per-Stream Scheduled Traffic (PSST) [1].

D. Per-Stream Filtering and Policing (PSFP)

Per-Stream Filtering and Policing (PSFP) is defined with the IEEE Std. 802.1Qci in 2017 to restrict the enqueueing of unwanted packets into the priority queues at the egress ports. Therefore it has the capability to drop packets or change the priority queue based on a combination of the time, the frame size, the rate and the burstiness of a stream. To accomplish that, PSFP consists of three layered components, the Stream Filters, the Stream Gates, and the Flow Meters. PSFP matches each arriving packet to the first Stream Filter instance where both the configured IEEE802.1CB Stream Handle and the vlan priority matches. The Stream Filter allows a wildcard entry for both the Stream Handle and the vlan priority. The matched Stream Filter instance may filter the frame based on the SDU Size, updates some counters, and defines the corresponding Stream Gate instance and the optional Flow Meter instance. The Stream Gate instance consists of a gate that can be opened and closed and assign an Internal Priority Value (IPV) based on a timing-based gate control list. When the packet arrives while the gate is in the closed state, the packet is dropped. Otherwise, the packet passes the gate and might change the switch's internal priority for that frame, based on the IPV value. In the following, we refer to this time-based gating mechanism as time-based PSFP.

The optional Flow Meter is a two rate three color token bucket meter as defined in MEF 10.3 [29]. The token bucket is configured using the committed information rate, committed burst size, excess information rate, and excess burst size as parameters. This mechanism enables to filter the frames based

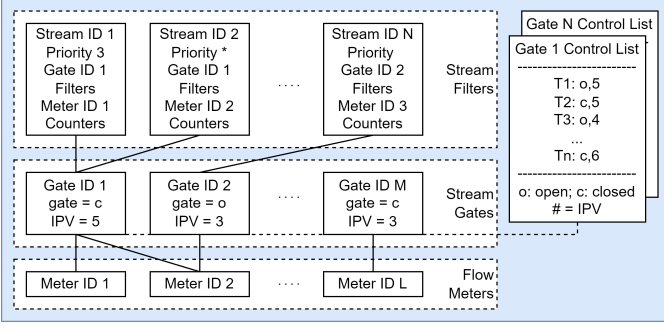


Fig. 2. Per-Stream Filtering and Policing

on their data-rate. In the following, we refer to this as rate-based PSFP.

The combination of those 3 layers enables a filtering of TSN packets based on their arrival time and data-rate.

IV. PROBLEM STATEMENT

PSST combines precise sending times at talkers (transmission offsets) and valid GCLs in the network. Misconfigured or malfunctioning end stations can disrupt the deployed network schedule by deviating from the planned sending behavior. We categorize frames that deviate from their planned arrival time, so-called faulty frames, into the following groups: missing frames, additional frames, early frames, and late frames. A missing frame is a frame that is scheduled to arrive at a specific time at a bridge but is never received by the bridge. The reason for missing frame might be a failed link, packet corruption (CRC failure), a broken end station, or an overloaded network component. An additional frame is a frame received by a bridge without being planned, which can be caused by a malfunctioning or misconfigured end station. Further, frames might arrive too early or too late at a bridge. Reasons for too early and too late frames are misconfigured end stations, synchronization errors, and non-deterministic queuing delay in a previous network component.

Faulty frames can cause a part of the network to no longer operate as planned or cause a global network failure. We explain and visualize the consequences of missing, additional, early, and late frames for a network-wide schedule in Section V.

V. EFFECTS OF FAULTY FRAMES IN SMALL EXAMPLES

In this section, we first introduce a method to visualize the behavior of an egress queue with TAS and periodic traffic. Then, we illustrate on a single link by minimal examples how faulty frames can delay later frames, possibly by significantly long time, and cause a persistent or even continuously increasing queue.

A. Visualization of Queue Behaviour with TAS and Periodic Traffic

In Figure 3 the occupancy of an egress queue is visualized over time. Frames of different streams are shown with their

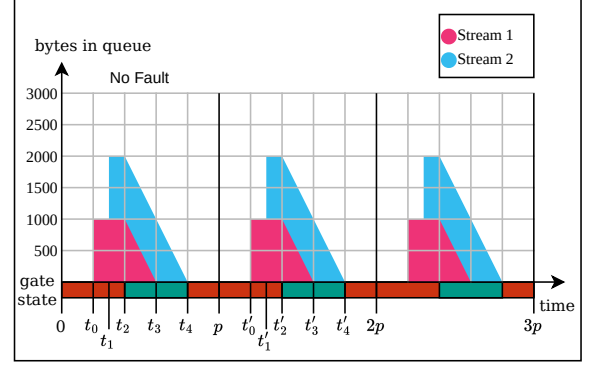


Fig. 3. Queue occupancy over time on an egress queue in the presence of TAS and periodic traffic w/o faulty frames. Sudden increases are frame arrivals, linear decreases indicate frame transmissions. Gate state: red means closed gate, green means open gate.

respective size in different colors on top of each other. In the example, the two frames are 1000 and 500 bytes large. Frames arrive periodically and instantaneously, which is indicated by a sudden increase of the queue occupancy. In contrast, we illustrate frame transmissions by linearly decreasing queue occupancy over time; the slope depends on the transmission speed¹. In the examples, all frames have the same period, and the beginning of a period is marked by a vertical line. In case of different periods, it is more useful to mark the start of a hyperperiod. The state of the periodic TAS gate is indicated on the x-axis. The gate is closed during red intervals, and it is open during green intervals. As a consequence, frames are queued during red intervals and transmitted during green intervals.

We briefly interpret the behavior illustrated in Figure 3. At times t_0 and t_1 the magenta and blue frames arrive with 1000 and 500 bytes, respectively. This can be seen from the sudden increases of the queue occupancy at t_0 and t_1 by these values. They are queued due to the closed gate. The gate opens at t_2 . This starts the transmission of the magenta frame, which is indicated by the continuously decreasing queue occupancy. After its completion at t_3 , the blue frame is transmitted until t_4 . The described behavior repeats due to the periodicity of the streams.

B. Effects of Faulty Frames on a Single Link

In the following, we illustrate the effects of faulty frames on a single link. The experiment setup is depicted in Figure 4. Possibly, multiple talkers send a stream to one listener via a single bridge. We consider the buffer occupancy on the link from the bridge to the listener. Periodic traffic is expected to arrive according to a schedule and sent within time slots controlled by the TAS. When the remaining time of the time

¹Instantaneous arrivals and continuous departures are inconsistent, but simplify the reading of the figures. When arrivals are also depicted continuous, simultaneous frame arrivals lead to superposed slopes, which looks more complex but does not add any value to the discussion.

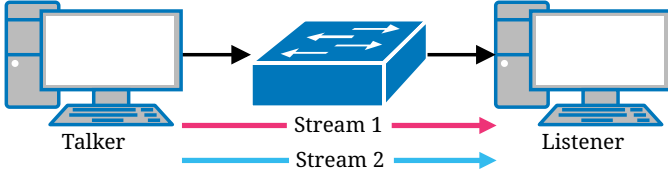


Fig. 4. When a single link is considered, multiple talkers send traffic over a single bridge to a listener. The queuing on the link from the bridge to the listener is considered.

slot is not large enough to accommodate the entire frame, the transmission of the frame is deferred.

We consider the impact of faulty frames, i.e., additional, late, early, or missing frames on potential delay of succeeding frames. Moreover, we show how unequal frame sizes can greatly increase that impact.

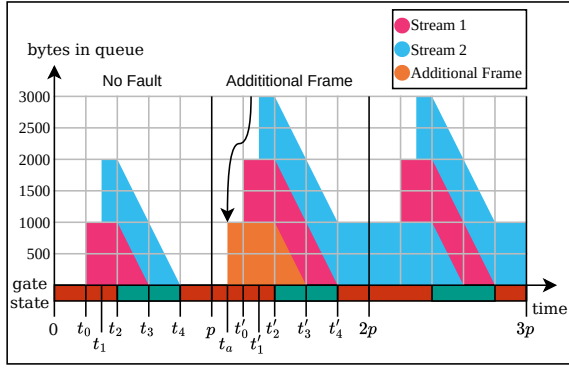


Fig. 5. In the second period an additional orange frame arrives before the magenta frame. As a consequence, the orange frame takes the time slot in that period and the magenta frame is significantly delayed. The time slot taken by the additional frame is missing in the future so that the queue will contain at least one frame and cause delay for all future frames.

1) *Impact of an Additional Frame:* Figure 5 visualizes the impact of an additional frame. A magenta frame and a blue frame are scheduled for every period. They arrive in time before their respective transmission slots and are queued for a short time. The gate opens slightly after their arrival so that the frames can be sent without significant delay. However, in the second period, an additional orange frame arrives at a time t_a before the arrival of the magenta frame. Therefore, the orange frame is transmitted when the gate opens, and the magenta frame remains queued until the gate re-opens again. The magenta frame then takes the transmission slot of the blue frame. When the blue frame arrives, it is queued until the gate re-opens next time, which is the transmission slot of the magenta frame. Thus, both the magenta and the blue frame miss their slots and are sent in transmission slot of other frames. A prerequisite for the latter is that there are at least two frames per period on the link.

For the following periods holds that whenever a frame arrives, there is already another frame in the queue so that the

old frame is sent while the new frame must wait again. This can also be viewed as follows. Normally, there are as many transmission slots as arriving frames. In case of additional frames, that number of frames is persistently buffered in the queue. As a consequence, all future frames will suffer from that queuing delay.

2) *Impact of a Late Frame:* Figure 6 visualizes the impact of a late frame. Here, a magenta and blue frame are scheduled for every period and normally arrive before two distinct transmission slots. Only in the second period, the magenta frame is late. As a consequence, the magenta frame cannot be sent within its time slot because the remaining time is not sufficient. Therefore, the frame is queued until the gate re-opens again. Thereby, the frame is delayed and takes the time slot of the blue frame, which misses its time slot.

In the next period, the effect of the delayed frame is like the effect of an additional frame: it takes the time slot of that period so that the frame arriving in that period is also significantly delayed, again. This can also be viewed as follows. If a frame is delayed and the time slot for that frame cannot be used, this time slot is missing in the future so that at least one frame remains in the queue. As a consequence, all future frames will suffer from that queuing delay.

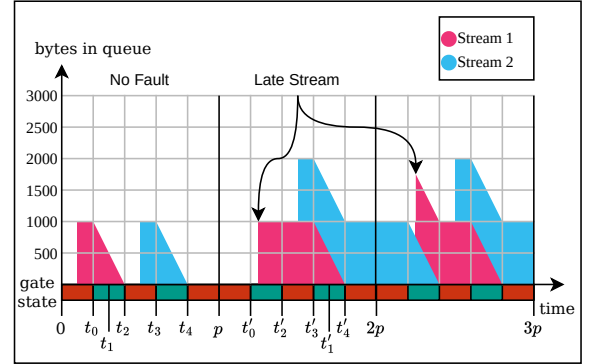


Fig. 6. The magenta stream is delayed so that all its frames miss their time slots. This effects that magenta and blue frames are significantly delayed and arrive at the next link in a wrong time slot.

3) *Impact of a Delayed Stream:* A stream is delayed if all its frames arrive late. Figure 6 visualizes the impact of a delayed stream. Here, a magenta and a blue frame are scheduled for every period, with distinct time slots that succeed their planned arrivals. However, all magenta frames arrive slightly too late so that they miss their time slot. As a consequence, they take the time slots of the blue frames that are also delayed for the time slots of the later magenta frames. Below the line, we observe that magenta and blue frames are transmitted in different time slots. That means, all frames are delayed, and they arrive at the next link in a wrong time slot.

4) *Impact of an Early Frame:* Figure 7 visualizes the impact of an early frame. Here, a magenta frame is scheduled and arrives before a blue frame so that it is transmitted first in the subsequent time slot. However, in the second period,

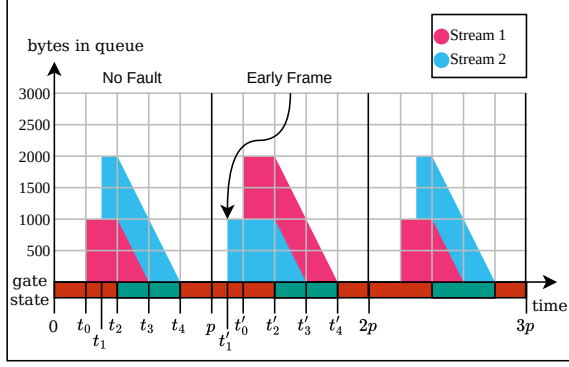


Fig. 7. A magenta frame usually arrives before a blue frame and is transmitted first when the gate opens. However, in the second period the blue frame arrives early and is transmitted first. Thereby, the magenta frame is sent slightly delayed. This may cause problems of a late frame on the next link (cf. Figure 6).

the blue frame is early and therefore transmitted first. As a consequence, the magenta frame is transmitted afterwards so that it is slightly late. As the magenta frame will arrive slightly delayed at the next link, it may cause the problems of a late frame on that link (cf. Section V-B2).

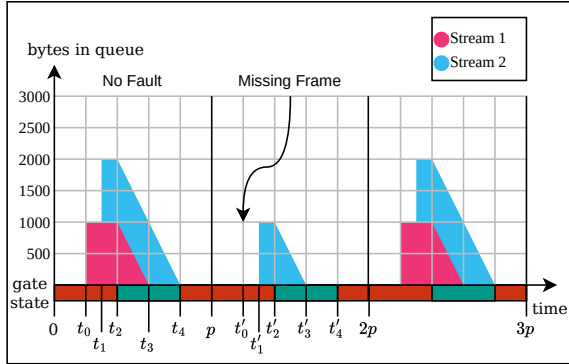


Fig. 8. In the second period the magenta frame is missing so that the blue frame is transmitted as soon as the gate opens. Thereby, the blue frame is sent slightly early. This may cause problems of an early frame on the next link (cf. Figure 7).

5) *Impact of a Missing Frame:* Figure 8 visualizes the impact of a missing frame. Here, a magenta frame is scheduled and arrives before a blue frame so that it is transmitted first in the subsequent time slot. However, in the second period the magenta frame is missing, and therefore the blue frame is transmitted as soon as the gates opens. As a consequence, the blue frame is transmitted earlier than planned. As the blue frame will arrive slightly early at the next link, it may cause the problems of an early frame on that link (cf. Section V-B4).

6) *Continuous Increase of Queue Occupancy in the Presence of Unequal Frame Sizes:* Figure 9 visualizes how a significantly delayed stream can cause a continuous increase

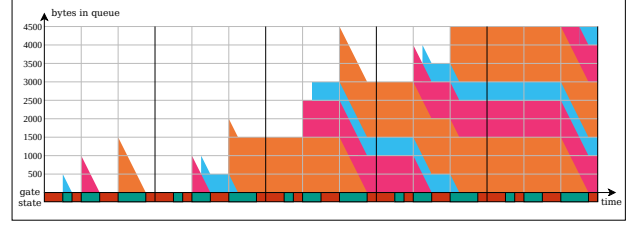


Fig. 9. The stream with small frame sizes is delayed, which changes the arrival order of the frame sizes relative to the sizes of the time slots. This leads to continuous increase of the queue occupancy.

of the queue occupancy in the presence of unequal frame sizes. We assume that three streams with small, medium, and large frames arrive within a period and their time slots succeed their frame arrivals. However, the stream with the small frames is delayed such that its frames arrive between the frames with medium and large frames sizes. This can happen due to a slightly delayed stream (cf. Section V-B3). We observe that when the small frame is sent in the time slot of the large frame, the succeeding large frame waits until the time slot for the next large frame. Thereby, two time slots remain unused, which increases the queue occupancy by two frames. As this process continuous, it leads to long frame delay and in the end to frame loss.

VI. SIMULATION AND VALIDATION

After showing the possible impacts of timing deviations in theory, this section simulates a small network using the network simulator OMNeT++ and its INET framework [30], [31]. The OMNeT++ simulations consist of three parts. First, the network topology file (Network.ned), which defines the end stations, TSN switches and the connections between them. Second, the OMNeT++ simulation initialization file (omnetpp.ini) that contains the configuration of all endstations and switches, including the GCL for all switches and the scheduled sending times for each stream. And third, a scenario file that controls the fault generations. To enable a stable fault generation, we had to apply a patch to the application generation, which we provide with all simulation files and instructions on GitHub. The simulations omit the gPTP time synchronization procedure to avoid interference between time synchronization and TSN applications, and prefill the MAC tables of all switches to prevent broadcast messages at the start of each simulation.

A. Validation of the Theoretical Results

To validate the theoretical results of Section V, all experiments were simulated with OMNeT++. The gate state over time as well as the transmission times were compared between the event log of the simulation and the above graphs. Both versions showed the same behavior. The source code for those simulations, together with a manual are published on GitHub (<https://github.com/EpplerM/TSN-Fault-Simulation>).

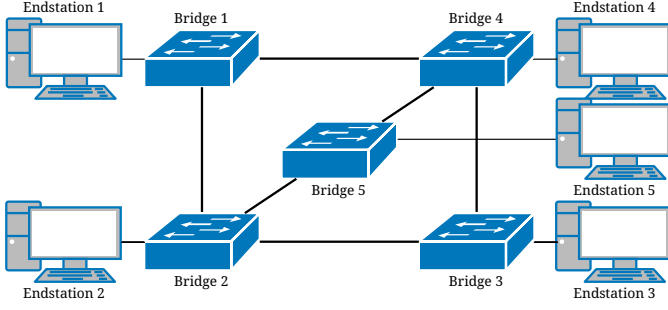


Fig. 10. Simulated network for Validation

B. Preliminary Simulation

Fig. 10 shows a simple network topology consisting of 5 bridges and 5 end-stations, connected with gigabit Ethernet links. Each bridge is connected to one end-station and two or three bridges. The end-stations send and receive scheduled streams, and the bridges use a TAS to forward them according to a predefined schedule. Table I shows the deployed streams as well as their scheduled path. The streams are sent periodically with a period length of $60 \mu s$ and a frame size between 250 and 1250 Bytes. While all streams are sent as scheduled,

TABLE I
STREAMS OF THE PROPAGATION EXAMPLE.

Stream	Talker	Path	Listener
A	ES 1	Bridge 1, Bridge 2, Bridge 3	ES 3
B	ES 2	Bridge 2, Bridge 3, Bridge 4	ES 4
C	ES 3	Bridge 3, Bridge 4, Bridge 1	ES 1
D	ES 4	Bridge 4, Bridge 1, Bridge 2	ES 2
E	ES 3	Bridge 3, Bridge 4, Bridge 5	ES 5
F	ES 4	Bridge 4, Bridge 5, Bridge 2	ES 2
G	ES 5	Bridge 5, Bridge 2, Bridge 1	ES 1

cally with a period length of $60 \mu s$ and a frame size between 250 and 1250 Bytes. While all streams are sent as scheduled,

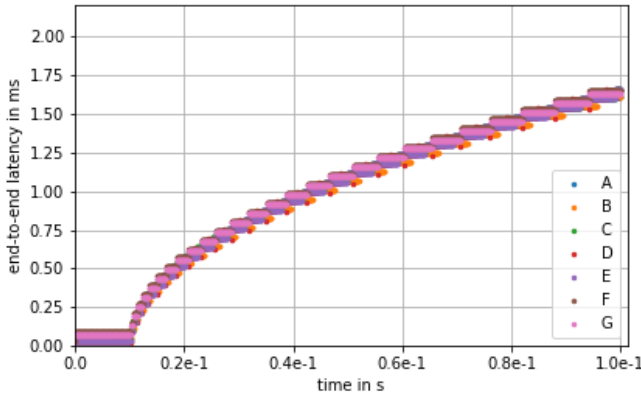


Fig. 11. End-to-End Latency of the applications A to G within the Network. After the fault at 10 ms, the latency of all frames increases over time.

all frames have a constant latency of less than $100 \mu s$.

C. Preliminary Results

A fault is introduced after 10 ms, by sending a single frame from stream A $10 \mu s$ later than scheduled. As a result, the frame remains in the queue in the same way as the example in Section V-B2. Fig. 11 shows an increasing delay for all streams, after the fault occurred. After less than 500 ms, all frames experience a delay of more than 10 times their regular delay. The simulation shows, that a system using PSST with the TAS, a delayed frame may lead to a fatal network error.

VII. CONCLUSION

PSST in TSN depends on the TAS and the sending time at the talkers. This results in a very fragile communication, as even a small problem may result in a fatal network error.

TSN scheduled traffic ist sehr fragile. Einzelnes Problem kann gesamten scheduled traffic lahmlegen. Folge ist Frabrikusfall. TSN scheduled traffic muss immer mit iwas abgesichert werden.

Naive ansätze wie hinzufügen von Slots für Fehler und vergrößern der Slots kann die Wahrscheinlichkeit für Fehler senken, das ursprüngliche Problem jedoch nicht lösen.

PSFP kann mit trivialem schedule die fehlerhaften frames dropen, was das Problem teilweise löst. In Kombination mit scheduling das Frame isolation implements, this solves the fragility.

TSN scheduled traffic needs Frame isolation and PSFP to ensure a working network.

Current Hardware is not capable of reliable PSFP filtering for each stream. This needs to be solved before TSN can be implemented in industrial environments.

In Zukunft muss methode gefunden werden um gegen Talking Idiots und Angreifer zu schützen. Wir vermuten, dass eine Kombination aus Frame Isolation und PSFP dies schaffen könnte.

The reliability of PSST in TSN strongly depends on the TAS configuration and the precise sending times of the talkers. Consequently, TSN scheduled traffic is inherently fragile – even minor timing deviations or configuration errors can lead to severe communication failures, potentially resulting in complete factory downtime.

Therefore, TSN scheduled traffic must always be safeguarded by additional mechanisms. The PSFP mechanism can mitigate part of this fragility by dropping faulty frames using a simple schedule. When combined with scheduling strategies that implement frame isolation, the overall robustness of TSN communication can be significantly improved. Thus, ensuring reliable TSN operation requires both PSFP and frame isolation.

However, most current hardware implementations are not yet capable of performing reliable PSFP filtering for every stream. This limitation must be resolved before TSN can be deployed in safety-critical industrial environments.

Future research should focus on developing methods to protect TSN networks from both unintentional misbehavior (“talking idiots”) and deliberate attacks. We hypothesize that a combination of frame isolation and PSFP could provide an effective foundation for such protection.

REFERENCES

- [1] "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)*, pp. 1–57, 2016.
- [2] T. Stüber, L. Osswald, S. Lindner, and M. Menth, "A survey of scheduling algorithms for the time-aware shaper in time-sensitive networking (tsn)," *IEEE Access*, vol. 11, pp. 61 192–61 233, 2023. DOI: 10.1109/ACCESS.2023.3286370
- [3] Z. Huang, H. Zhu, H. Zhang, and T. Huang, "A Scalable Heuristic Time-Sensitive Traffic Scheduling Algorithm for In-Vehicle Network," in *International Conference on Hot Information-Centric Networking (HotICN)*, 2022, pp. 111–118.
- [4] S. S. Craciunas and R. S. Oliver, "Out-of-Sync Schedule Robustness for Time-sensitive Networks," in *IEEE International Conference on Factory Communication Systems (WFCS)*, 2021, pp. 75–82.
- [5] Z. Feng, M. Cai, and Q. Deng, "An Efficient Pro-Active Fault-Tolerance Scheduling of IEEE 802.1Qbv Time-Sensitive Network," *IEEE Internet of Things Journal*, vol. 9, no. 16, pp. 14 501–14 510, 2021.
- [6] A. A. Atallah, G. B. Hamad, and O. A. Mohamed, "Routing and Scheduling of Time-Triggered Traffic in Time-Sensitive Networks," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 7, pp. 4525–4534, 2020.
- [7] N. Reusch, P. Pop, and S. S. Craciunas, "Work-In-Progress: Safe and Secure Configuration Synthesis for TSN Using Constraint Programming," in *IEEE Real-Time Systems Symposium (RTSS)*, 2020, pp. 387–390.
- [8] V. Gavrilut, B. Zarrin, P. Pop, and S. Samii, "Fault-Tolerant Topology and Routing Synthesis for IEEE Time-Sensitive Networking," in *International Conference on Real-Time Networks and Systems (RTNS)*, 2017, pp. 267–276.
- [9] Z. Feng, Q. Deng, M. Cai, and J. Li, "Efficient Reservation-based Fault-Tolerant Scheduling for IEEE 802.1Qbv Time-Sensitive Networking," *Journal of Systems Architecture*, vol. 123, 2022.
- [10] R. Dobrin, N. Desai, and S. Punnekkat, "On Fault-Tolerant Scheduling of Time Sensitive Networks," in *International Workshop on Security and Dependability of Critical Embedded Real-Time Systems (CERTS)*, vol. 73, 2019, 5:1–5:13.
- [11] Z. Pang et al., "Flow Scheduling for Conflict-Free Network Updates in Time-Sensitive Software-Defined Networks," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 3, pp. 1668–1678, 2021.
- [12] A. A. Atallah, G. B. Hamad, and O. A. Mohamed, "Fault-Resilient Topology Planning and Traffic Configuration for IEEE 802.1Qbv TSN Networks," in *IEEE International Symposium on On-Line Testing And Robust System Design (IOLTS)*, 2018, pp. 151–156.
- [13] Y. Zhou, S. Samii, P. Eles, and Z. Peng, "Reliability-Aware Scheduling and Routing for Messages in Time-Sensitive Networking," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 20, no. 5, pp. 1–24, 2021.
- [14] A. A. Syed, S. Ayaz, T. Leinmüller, and M. Chandra, "Fault-Tolerant Static Scheduling and Routing for In-vehicle Networks," in *International Telecommunication Networks and Applications Conference (ITNAC)*, 2022, pp. 273–279.
- [15] Y. Zhou, S. Samii, P. Eles, and Z. Peng, "ASIL-Decomposition Based Routing and Scheduling in Safety-Critical Time-Sensitive Networking," in *IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2021, pp. 184–195.
- [16] H. Li, H. Cheng, and L. Yang, "Reliable Routing and Scheduling in Time-Sensitive Networks," in *IEEE International Conference on Mobility, Sensing and Networking (MSN)*, 2021, pp. 806–811.
- [17] A. A. Syed, S. Ayaz, T. Leinmüller, and M. Chandra, "Fault-Tolerant Dynamic Scheduling and Routing for TSN based In-vehicle Networks," in *IEEE Vehicular Networking Conference (VNC)*, 2021, pp. 72–75.
- [18] A. A. Syed, S. Ayaz, T. Leinmüller, and M. Chandra, "Dynamic Scheduling and Routing for TSN Based In-vehicle Networks," in *IEEE International Conference on Communications Workshops (ICC Workshops)*, 2021, pp. 1–6.
- [19] "IEEE Standard for Local and metropolitan area networks–Frame Replication and Elimination for Reliability," *IEEE Std 802.1CB-2017*, pp. 1–102, 2017.
- [20] A. A. Syed, S. Ayaz, T. Leinmüller, and M. Chandra, "Network Coding Based Fault-Tolerant Dynamic Scheduling and Routing for In-Vehicle Networks," *Journal of Network and Systems Management*, vol. 31, no. 1, p. 27, 2023.
- [21] F. Pozo, G. Rodriguez-Navas, and H. Hansson, "Schedule Reparability: Enhancing Time-Triggered Network Recovery Upon Link Failures," in *International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, 2018, pp. 147–156.
- [22] S. S. Craciunas, R. S. Oliver, M. Chmelík, and W. Steiner, "Scheduling Real-Time Communication in IEEE 802.1Qbv Time Sensitive Networks," in *International Conference on Real-Time Networks and Systems (RTNS)*, 2016, pp. 183–192.
- [23] M. Vlk, Z. Hanzálek, K. Brejchová, S. Tang, S. Bhattacharjee, and S. Fu, "Enhancing Schedulability and Throughput of Time-Triggered Traffic in IEEE 802.1Qbv Time-Sensitive Networks," *IEEE Transactions on Communications*, vol. 68, no. 11, pp. 7023–7038, 2020.
- [24] "IEEE Standard for Local and Metropolitan Area Networks - Virtual Bridged Local Area Networks Amendment 12: Forwarding and Queuing Enhancements for Time-Sensitive Streams," *IEEE Std 802.1Qav-2009 (Amendment to IEEE Std 802.1Q-2005)*, pp. C1–72, 2010.
- [25] "IEEE Standard for Local and metropolitan area networks–Bridges and Bridged Networks–Amendment 29: Cyclic Queuing and Forwarding," *IEEE 802.1Qch-2017 (Amendment to IEEE Std 802.1Q-2014 as amended by IEEE Std 802.1Qca-2015, IEEE Std 802.1Qcd(TM)-2015, IEEE Std 802.1Q-2014/Cor 1-2015, IEEE Std 802.1Qbv-2015, IEEE Std 802.1Qbu-2016, IEEE Std 802.1Qbz-2016, and IEEE Std 802.1Qci-2017)*, pp. 1–30, 2017.
- [26] "IEEE Standard for Local and Metropolitan Area Networks–Bridges and Bridged Networks - Amendment 34: Asynchronous Traffic Shaping," *IEEE Std 802.1Qcr-2020 (Amendment to IEEE Std 802.1Q-2018 as amended by IEEE Std 802.1Qcp-2018, IEEE Std 802.1Qcc-2018, IEEE Std 802.1Qcy-2019, and IEEE Std 802.1Qcx-2020)*, pp. 1–151, 2020.
- [27] O. Gabbard and P. Kaul, "Time-division multiple access," in *EASCON'74; Electronics and Aerospace Systems Convention*, 1974, pp. 179–184.
- [28] "IEEE Standard for Local and Metropolitan Area Networks–Timing and Synchronization for Time-Sensitive Applications," *IEEE Std 802.1AS-2020 (Revision of IEEE Std 802.1AS-2011)*, pp. 1–421, 2020.
- [29] M. Standard, "Mef 10.3 ethernet servicer attributes phase 3," 2013.
- [30] A. Varga, "OMNeT++," in *Modeling and Tools for Network Simulation*, K. Wehrle, M. Güneş, and J. Gross, Eds. Springer Berlin Heidelberg, 2010, pp. 35–59.
- [31] L. Mészáros, A. Varga, and M. Kirsche, "Inet framework," in *Recent Advances in Network Simulation: The OMNeT++ Environment and its Ecosystem*, A. Virdis and M. Kirsche, Eds. Cham: Springer International Publishing, 2019, pp. 55–106, ISBN: 978-3-030-12842-5. DOI: 10.1007/978-3-030-

12842-5_2 [Online]. Available: https://doi.org/10.1007/978-3-030-12842-5_2