2019 International Conference on Networked Systems (NetSys)

A Simulation of Asynchronous Traffic Shapers in Switched Ethernet Networks

Alexej Grigorjew
University of Würzburg, Germany
alexej.grigorjew@uni-wuerzburg.de

Florian Metzger
University of Würzburg, Germany
florian.metzger@uni-wuerzburg.de

Tobias Hoßfeld
University of Würzburg, Germany
tobias.hossfeld@uni-wuerzburg.de

Johannes Specht
University of Duisburg-Essen, Germany
johannes.specht@uni-due.de

***Abstract*—As the requirements of real-time networking increase, the IEEE Time-Sensitive Networking (TSN) task group is working on new standards for high bandwidth and low latency Ethernet switching. In particular, the upcoming P802.1Qcr standard features Asynchronous Traffic Shaping (ATS), which is a promising approach for real-time networks with high dynamics and critical safety requirements. Additionally, parallel concepts such as frame preemption can be used to reduce delays even further.**

This work presents a discrete event simulation framework that can analyze the performance of these concepts. It implements frame preemption and multiple shaping algorithms, while recording latency and queue utilization statistics at the switches. In addition, an evaluation of a linear topology with four switches is performed under high link utilization, which demonstrates the benefit of the above mechanisms compared to regular Ethernet switching.

***Index Terms*—Asynchronous Traffic Shaping, Time-Sensitive Networks, Real-Time Networks, Latency, Simulation, Frame Preemption**

I. Introduction

Real-time communication systems have a long history both in multimedia communication as well as industrial and automation networks. While the latter often require zero packet loss and deterministic bounds regarding the worst-case latency of time-sensitive streams, many multimedia applications can be satisfied by statistical Quality of Service (QoS) guarantees when paired with high link utilization. Recent developments in production and automotive systems reveal these demands side by side. On the one hand, the network must support massive amounts of sensors, actors, and control units with deterministic requirements down to a few microseconds [1]. On the other hand, cameras, control interfaces, and even third party applications are possibly using the same network infrastructure, requiring high bandwidth and exhibiting irregular traffic patterns.

Historically, such an effort required the use of dedicated communication channels. Popular approaches for simple machine-to-machine communication are bus protocols, such as CAN [2], CAN-FD [3], Profibus [4], and FlexRay [5]. However, they suffer from low bandwidths around 1–10 Mbit/s and small frame sizes, which makes it difficult to encode, or even verify, message origins in dynamic environments. Therefore, various projects enhance the regular Ethernet technology by real-time capabilities to enable its use in the industrial and automotive context, like Profinet [4], [6] and EtherCAT [4]. Similarly, the IEEE 802.1 Time-Sensitive Networking (TSN) task group [7] is working on extensions and next generation features to the existing Ethernet standards, featuring real-time networking at 100 Mbit/s and higher [8], as required by the increasing amount of sensors, radars, and cameras in modern appliances.

Initially developed in the context of multimedia transmissions, the standard Audio Video Bridging (AVB) Systems [8] defines general requirements and latency profiles for asynchronous real-time switching. The original system divides traffic into multiple classes and combines Credit-Based Shaping [9] on a per-class basis with priority based transmission selection at each hop. This prevents high priority classes from using more bandwidth than they allocated and gives an upper bound on the latency they cause. However, as all streams of a class share the same aggregate credit counter, interfering traffic could still lead to accumulating delays at high utilization [10]. The TSN group has since developed several extensions to real-time Ethernet. In particular, IEEE 802.1AS [11] and IEEE 802.1Qbv [12] specify time synchronization and scheduling mechanisms that enable extremely low and predictable end-to-end latencies. However, they focus on periodic traffic, require network-wide planning, rely on clock synchronization, and often sacrifice available bandwidth for lower latency.

As an alternative, IEEE P802.1Qcr [13] describes per-stream Asynchronous Traffic Shaping (ATS) extensions. ATS provides per-hop latency bounds, even when applied with per-priority interleaved queuing [14]. This is particularly interesting in dynamic or safety-critical environments, where the clock synchronization would pose an additional single point of failure. In addition, to prevent large ongoing low-priority transmissions from delaying higher priorities, frame preemption [15], [16] can be applied with a select subset of

978-1-7281-0568-0/19/$31.00

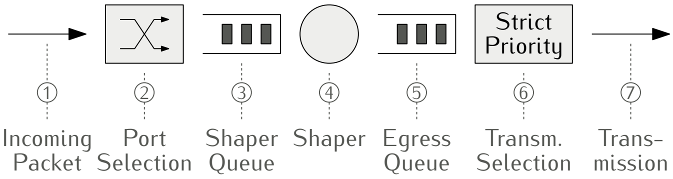


Figure 1. Overview of the processing of frames in a single switch.

priorities.

Although some analytical bounds for ATS are available [14], [17], [18], the effects of frame preemption on different priority levels are hardly considered. Besides, most analytic models in the context of real-time networks focus on the worst-case latency, but the stochastic properties, especially of lower priorities, remain rather unexplored. For instance, since unused bandwidth is available to lower priorities in ATS, the actual amount of best effort traffic that can be transmitted alongside real-time frames poses an interesting question.

Contribution: This work describes the design of our discrete event simulation that implements the data plane behavior of ATS switches, including different shaping algorithms and enhanced by frame preemption. Furthermore, it shows an example evaluation of a linear topology with 4 switches and varying interference, providing initial insights on the effects of frame preemption on high-priority streams.

The remainder of this work is structured as follows. Section II explains the design and important implementation aspects of the simulator. The evaluated topology is described in Section III, whereas the results of its simulation are presented in Section III-A. Section IV includes a discussion of the approach as well as interesting extensions and applications in future works. Finally, Section V summarizes the contribution and concludes the paper.

II. Simulation Design

The system behavior under investigation can be summarized as an extension of a priority scheduled Ethernet switch with additional shaping algorithms and carefully allocated queues [13], [14]. The idea is to smooth out bursts of individual streams and prevent accumulating latencies and interference by regulating each stream's sending rate at each hop, providing deterministic delay bounds as long as every stream remains within its specified traffic characteristics.

The concept is implemented as a discrete event simulation, while the model itself is programmed as process oriented simulation in SimPy [19] and converted into events by the framework. For each stream in the simulated scenario, there is an arrival process which determines the interarrival time between two consecutive frames. These processes are started directly at the beginning of the simulation, in parallel. Each time an arrival process generates a frame for its stream, it is repeatedly transmitted towards its next hop on the stream's path until it is received by the listener. For each such individual transmission, the shaping delays, queueing delays, and transmission delays are recorded separately.

Fig. 1 provides a high level overview of a single transmission process at every hop and for every frame. At first, each incoming packet (1) is analyzed and its resources are prepared, e.g., the next hop, the used link, and the queues and shaper states. Thereby, the egress port towards the next hop is selected (2). All of the following queues, states and resources are bound to this egress port. If shaping is enabled, the packet arrives in the shaper queue (3). In the non-interleaved case, there is exactly one shaper queue for each individual stream in the simulation. For the realistic, interleaved model, there is one shaper queue for each other port and for each priority level at this egress port. This work is focused on the so called Token Bucket Emulation (TBE) [14], [20] interleaved shaping algorithm (4). The alternative algorithm considered by scientific literature so far is Length Rate Queuing (LRQ) [14], [21], [22]. This is in line with IEEE P802.1Qcr [13], which specifies a timestamp based algorithm equivalent to TBE.

With TBE, each shaper instance contains the current number of tokens n_{tokens} for each individual stream. At each event involving the shaper state, i.e., when enqueuing new frames and dequeing the frame at the top of the queue, this number is updated based on the shaping algorithm and the elapsed time since the last update. Each stream has its own bucket size b and token generation rate r, and the new shaper state is calculated as follows.

$$n_{tokens} = \min\left\{b,\ n_{tokens} + (t_{now} - t_{last\_update}) \cdot r\right\} \qquad (1)$$

As soon as there are sufficient tokens available to transmit the first packet in the shaper queue, it is moved towards the egress queue (5) and tokens are subtracted based on the frame's size. Here, there is exactly one egress queue for each supported priority level at this respective egress port. The strict priority transmission selection (6) now decides when to transmit the first packet in the queue. This is where most interference from other streams delays the frame: the ongoing transmission must be finished (or preempted), all frames from the same or higher priority streams that were already enqueued are transmitted first, and all frames from higher priorities that arrive during this time take precedence. The transmission process (7) is based on the link speed and the size of the frame. In addition, 20 extra bytes are considered for the preamble and the interpacket gap.

A. Frame preemption

When preemption is enabled, the transmission of a lower priority frame can be paused in favor of a higher priority. Thereby, an additional preamble, interpacket gap and intermediate checksum are transmitted. Note that our model supports preemption accuracy, i.e., a transmission can only be interrupted at steps of 64 bytes, and only when both parts of the preempted frame are at least 64 bytes long (minimum Ethernet frame size) [15], [16]. In the simulation, an event is scheduled at the moment when the next multiple of 64 bytes is transmitted. It transmits the additional 4 bytes for the intermediate checksum, waits 12 byte times for the interpacket gap (IPG), starts the transmission process of the higher prioritized

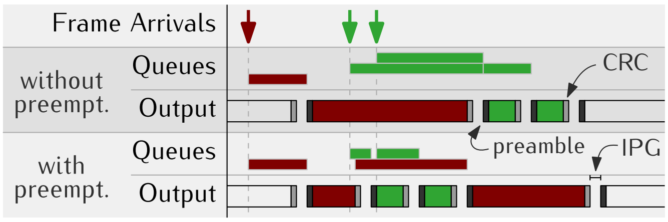


Figure 2. Preemption of a low-priority (red) frame by two incoming high-priority (green) frames. Each preemption comes with an overhead of one preamble, checksum (CRC) and interpacket gap (IPG).

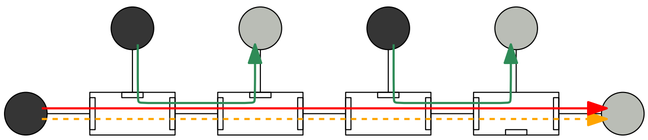


Figure 3. Evaluated topology. Red: measured periodic high-priority streams. Green: periodic high-priority interference streams. Orange/dashed: exponential low-priority interference. Each type of stream measures 0.4 Gbit/s.

frame, and re-enqueues the remainder of the preempted frame at the top of its respective queue. Only one frame can be preempted at a time, but multiple high-priority packets can be transmitted before the preempted frame is resumed. In addition, after resumption, the same frame can be preempted once again, which complicates its formal delay analysis.

Fig. 2 illustrates an example switch output with and without frame preemption, sending one large low-priority frame (red) followed by two small high-priority frames (green). Without preemption, both high-priority frames must wait for the full remaining duration of the red packet transmission. If preemption is applied, the transmission of the green packet can begin almost immediately, after finishing the current red 64 bytes block, an intermediate checksum (CRC), and the interpacket gap. Note that the resumed transmission of the red packet finishes later than the green packet in the regular case, since the packet is not simply split, but there is an overhead of one additional CRC (4 bytes), preamble (8 bytes), and IPG (12 bytes) on the link.

B. Simulation Output

The simulator reports its results in two ways. One output file contains a detailed log of all packet transmissions. For every each individual hop, the shaper delay (3+4), queuing delay (5+6) and transmission delay (7) is reported separately. A second output file keeps track of the used queuing resources of each switch in the network. Every time a packet enters or leaves one of the queues (shaper and egress), both the number of currently enqueued packets as well as their size in bytes is reported with the current timestamp. These numbers are reported locally, for the respective queue, as well as globally, i.e., the sum of all queues in the current switch.

III. Evaluation

Fig. 3 provides an overview of the evaluated topology. In this initial work, the behavior of high priority (HP) streams is investigated which are interfering with other HP streams as well as low priority best effort (BE) traffic. The topology shows four switches in a row, three talker nodes (dark gray) and three listener nodes (light gray). The path of the investigated HP streams is shown in red, the other interfering HP streams are shown in green, and the BE interference is displayed as a dashed, orange line, using the same path as the red streams. Each of these three paths consists of 20 individual streams with a total data rate of 0.4 Gbit/s. All links in the topology provide a bandwidth of 1 Gbit/s, which means that the two links that carry all three types of streams are overprovisioned at 120% utilization. This topology is inspired by the higher levels of a hierarchical topology connecting branches of multiple sensors, which are abstracted by a few talker nodes here.

The static frame size ℓ_s of each HP stream s is determined randomly between 64 bytes and 1522 bytes at the start of the simulation. Together, 20 streams carry a data rate of 0.4 Gbit/s. The data rate was distributed evenly among all streams, i.e., each individual HP stream s features a data rate of $r_s = 20$ Mbit/s. The deterministic interarrival time t_s of the packets is then given by $t_s = \ell_s / r_s$. This periodic and deterministic behavior is inspired by simple machine-to-machine communication patterns.

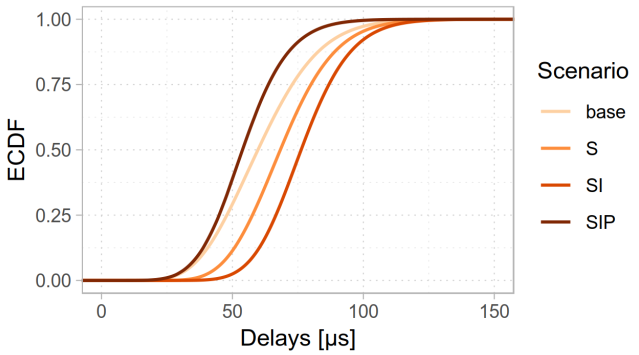
The interarrival times of BE frames are exponentially distributed. 70% of these packets have the maximum frame size of 1522 bytes, the remainder has a small packet size of 128 bytes. The minimum interarrival time is 1 µs, and the mean interarrival time is chosen to reflect a mean throughput of 20 Mbit/s for each BE stream. These packet sizes are chosen to resemble common web communications, such as HTTP requests (128 byte packets) and downloads (1522 byte packets), while the exponential interarrival times are an approximation of the superposition of independently distributed flows, as suggested by the Palm-Khintchine Theorem [23].

In this evaluation, four scenarios are considered. The baseline scenario (*base*) represents regular Ethernet switching with priority-based transmission selection, but without shaping or preemption. The shaped scenario (*S*) applies TBE shaping as described in Section II with separate shaper queues for each single stream in the network. The interleaved scenario (*SI*) applies interleaved shaping with one common shaper queue for each possible ingress and priority at each egress port. Finally, the last scenario applies preemption (*SIP*) in addition to interleaved shaping in an attempt to lower the latency caused by interfering BE traffic.

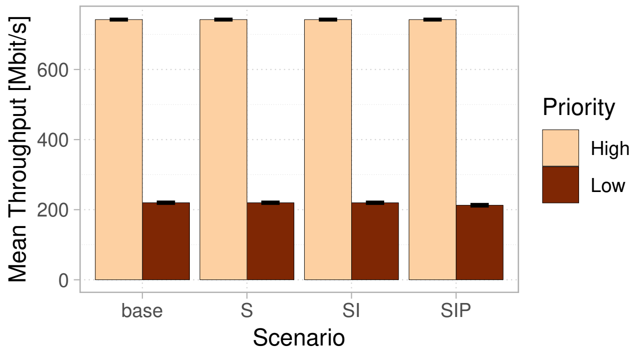
The reported delay values in the evaluation results refer to the end-to-end path from the talker towards the receiver. These measured delays are the sum of the waiting times in the shaper queue (3+4), the egress queue (5+6), and the transmission time (7), as indicated in Fig. 1.

A. Evaluation Results

With the basic scenario setup and methodology in mind, we can now examine the results in the following paragraphs. All simulations cover a duration of 30 seconds and all scenarios



(a) Empirical cumulative distribution function (ECDF) of the measured end-to-end delays of high-priority streams during one simulation run.



(b) Mean throughput with 95% confidence intervals from 10 runs.

Figure 4. Empirical distribution of measured delays and mean throughput; with and without shaping (S), interleaving (I), and preemption (P); with shaping, delays increase slightly, while their variance decreases and throughput remains unchanged.

were repeated 10 times. Where applicable, either the maximum or mean results with a 95 % confidence interval are given.

Overview: For a general overview, Fig. 4a shows the empirical cumulative distribution function of all measured end-to-end delays from one of the 10 simulation runs. As can be expected when introducing additional queuing in the system, enabling shaping (*S*) and interleaving (*SI*) generally increases the measurable delay of the real-time flows compared to the *base* scenario. However, at the same time, the curves are steeper, which suggests a lower variance and a more predictable behavior. In fact, despite the increase in the average case, the latency is now bounded, which is the main incentive of TSN and not directly apparent in this figure.

Applying frame preemption in addition to shaping (*SIP*) reduces the mean delay of the HP streams by a noticeable margin, even below the original non-shaped scenario. Compared to the other curves, the variance can be reduced even further thereby. Note, however, that this representation does not show the negative impacts on the latency of the lower priority BE traffic. As the links are congested in this scenario, the latency of the BE frames is mainly dependent on the configured size of the queues, and therefore omitted here.

Nonetheless, the congested scenario is well suited for a comparison of actually attained throughputs in all of the scenarios. Therefore, Fig. 4b shows the mean attained goodput (without preamble, CRC and IPG) with 95% confidence intervals for each configuration. While the majority of these values are nearly identical, only low priority BE traffic shows a small penalty with preemption (*SIP*). This is due to the additional overhead of preamble and IPG, as mentioned in Section II-A.

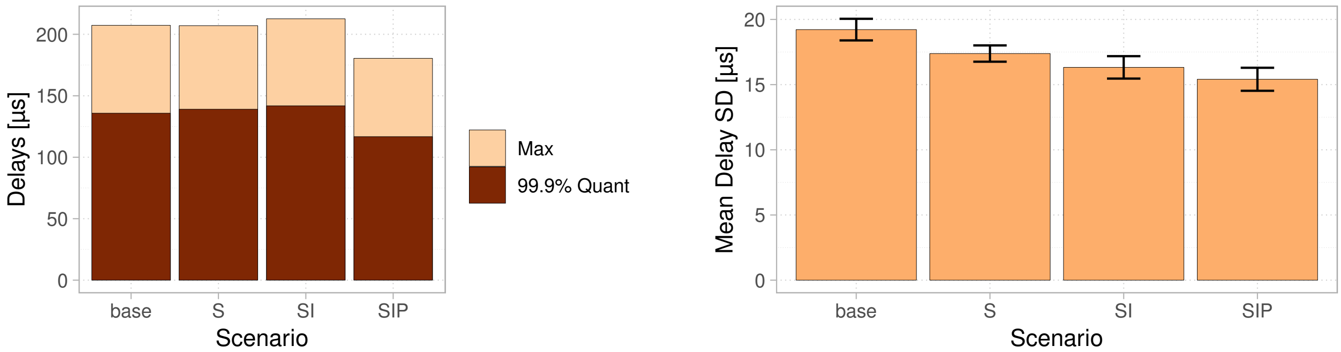
Quantiles and Variance: Taking a closer look at the delays, Fig. 5a shows their observed worst cases and 99.9% quantiles throughout all 10 simulation runs. Unlike the general distribution of delays, the observed worst-case delays hardly change when enabling traffic shaping. Only interleaving (*SI*) causes a slight increase to both the maximum and the 99.9% quantile. In contrast, preemption (*SIP*) noticeably reduces the worst-case delay. In theory, the worst-case bound would be lowered by the difference between the largest BE frame size (1522 bytes) and the largest un-preemptable Ethernet frame (127 bytes).

Note that the simulated results only apply to the considered scenario and do not represent the absolute attainable worst-case due to the stochastic nature of the simulation. However, the simulation can be used to reconstruct theoretical worst-cases and investigate the tightness of analytical bounds. Nevertheless, for use cases that do not require strict deterministic bounds, such as multimedia applications and interactive communication systems, these practical results are particularly valuable. For such investigations, quantiles are also an important factor: by ignoring the top 0.1% of the samples, the delay is lowered by more than 25%.

As variance is an important factor for the perceived quality of a connection, Fig. 5b displays the mean standard deviations of the delays with 95% confidence intervals. Like indicated earlier, despite increasing the mean delay, shaping actually reduces the variance of the streams. The figure shows a statistically significant difference of standard deviation between the *base* scenario and interleaved shaping (*SI*). Adding preemption shows a trend towards even lower variance here.

Interestingly, a beneficial impact of interleaved shaping (*SI*) over classic shaping with per-flow queues (*S*) can be observed. That is, interleaved shaping reduces delay variation, while it comes at a better implementation complexity than classic shaping. However, as the confidence intervals overlap, further tests are required to confirm this trend in the future.

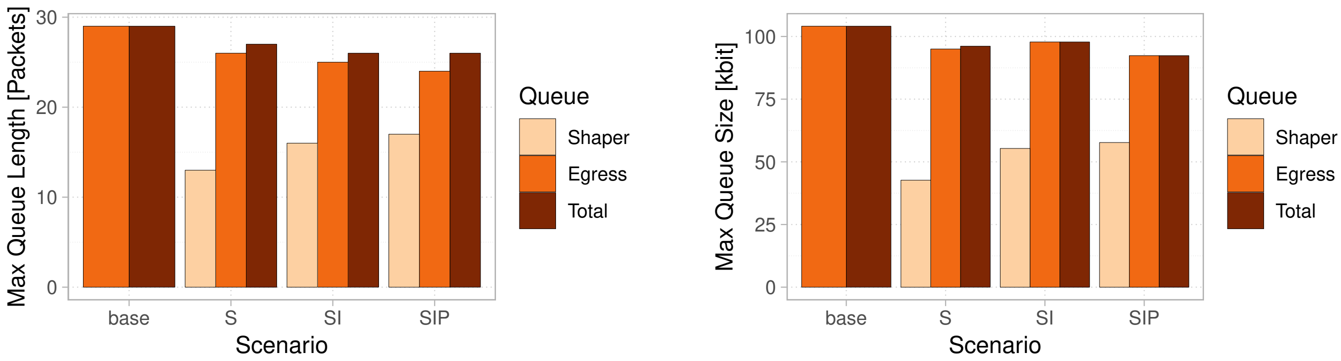
Queue Sizes: As mentioned in Section II-B, the simulation can not only provide insights on the delays of ATS enabled switches. Fig. 6 displays the maximum observed queue lengths (number of frames) and sizes (in kbit) for HP frames in all switches during all 10 simulation runs. While the queue utilization can be investigated separately for each queue in the model, many hardware implementations apply a common memory architecture where the actual frame data resides in the same data structure. Therefore, the *total* queue sizes of each device are of particular importance for



(a) Maximum delays and their 99.9% quantiles obtained from all 10 runs.

(b) Mean standard deviation with 95% confidence intervals from 10 runs.

Figure 5. Measured maximum delays and their standard deviations; with and without shaping (S), interleaving (I), and preemption (P).



(a) Maximum number of queued high-priority frames.

(b) Maximum size of queued high-priority frames.

Figure 6. Maximum observed queue sizes for HP streams in all switches during all 10 simulation runs; with and without shaping (S), interleaving (I), and preemption (P).

hardware manufacturers, stream reservation procedures and network planning in general. Despite increasing the *mean* delay of frames, Fig. 6 suggests that the *maximum* amount of simultaneously queued packets can be reduced by shaping.

Note that, once again due to the overprovisioned links in this scenario, the figures cannot show possible negative implications on the queue sizes of low priority traffic. However, keep in mind that BE traffic is allowed to be dropped either way. Furthermore, if zero packet loss is a critical requirement, proven analytical bounds such as [18] can be employed during stream reservation. Nevertheless, knowledge about expected typical resource usage can be applied during hardware design when both critical high-priority traffic and lossy BE traffic are present in the same network.

IV. Discussion

The above evaluation refers to a simple scenario that can only grasp a limited view on the effects of ATS. It provides a general overview of the possibilities and system behavior while also introducing parameters and metrics that have not been investigated in this context before, most notably the effects of preemption and the consideration of quantiles for relaxed real time requirements. These observations open a broad field of further research questions for future works in this direction, and some of them are presented here.

Frame preemption is an interesting topic, especially when small high priority packets are served next to best effort traffic. In extreme cases, delay bounds smaller than 10 µs may be required. Consider a high priority packet arriving right after the transmission of a maximum sized best effort frame has started. With a 1 Gbit/s link, the store and forward operation for 1542 bytes already requires more than 12 µs, effectively denying the service goal on its own. When using preemption, the biggest non-preemtible frame size is 127 bytes, which lowers the impact on the high priority delay to a maximum of 1 µs. In addition, when more priorities are considered, the negative impacts of frame preemption on low priority latencies should be investigated. A single frame can be preempted multiple times, and during each preemption, multiple high priority frames can be transmitted. The assignment of preemtable and preempting priority levels will be a key consideration for strict real-time environments.

A separate investigation of analytical methods for statistical evaluation could provide a complementary contribution and create an opportunity for comparison and verification in both directions. On a similar note, all attained findings based on simulation results are only applicable for the investigated scenarios. While general statements are hard to obtain, the relations between the configuration of streams and their analytic bounds is an important topic for network dimensioning. Re-enacting worst-case configurations in a simulation provides an impression of the tightness of delay bounds. Similarly, simulation of typical use cases can yield guidelines for queue dimensioning under different stream configurations.

With the above applications in mind, the simulation framework can be extended in several ways. For example, data allocation in switches is usually organized in blocks rather than pure bytes. This can be an important addition when taking a closer look at queue sizes. Similarly, processing delays in switches are currently not considered and may be added at an appropriate abstraction level.

Finally, as mentioned in Section I, the TSN group is working on multiple approaches for real-time networking. Hence, it would be worthwhile to implement the Credit-based Shaper Algorithm [9] and the enhancements for scheduled traffic [12] in the simulation. A numerical comparison of the achievable delay and throughput in typical scenarios could deliver valuable results, especially with regard to soft real-time criteria — such as delay quantiles slightly below the maximum value as well as a low delay variation.

V. Conclusion

Latency and bandwidth requirements in time-sensitive networks continue to grow, and ongoing standardization efforts strive to meet these demands. This work presents a novel simulation framework, featuring IEEE P802.1Qcr Asynchronous Traffic Shaping and IEEE 802.1Qbu frame preemption, that aids at analyzing the performance of these networks and dimensioning switch resources.

An initial evaluation of a linear network with four switches and high link utilization reveals some benefits of these approaches. Per stream traffic shaping with interleaved queuing provides deterministic delay bounds, while increasing the mean latency by a small amount and still achieving full link throughput. At the same time, delay variance can actually be decreased by interleaved shaping. Frame preemption can reduce high priority frame delays even below the baseline scenario and decreases delay variance even further.

In the future, it will be interesting to investigate the delay characteristics of various network topologies and scenarios in relation to the analytical bounds from literature, especially regarding the tightness of the bounds in practical situations. In addition, the simulation can be extended in several ways, including more detailed latency and resource models, and new time-sensitive networking approaches. Finally, frame preemption can also impact latency of preempted low priority streams negatively. The extend of these effects requires further analysis.

References

[1] E. Grossman. Deterministic Networking Use Cases. draft-ietf-detnet-use-cases-19 (Informational), October 2018.

[2] ISO 11898. Road Vehicles — Controller Area Network (CAN). *International Organization for Standardization*, 2015.

[3] Florian Hartwich (Robert Bosch GmbH). CAN with Flexible Data-Rate. In *Proc. iCC*, pages 1–9, 2012.

[4] IEC 61158. Industrial communication networks - Fieldbus specifications. *International Electrotechnical Commission*, 2014.

[5] ISO 17458. Road Vehicles — FlexRay communications system. *International Organization for Standardization*, 2013.

[6] IEC 61784-2. Industrial communication networks - Profiles - Part 2: Additional fieldbus profiles for real-time networks based on ISO/IEC 8802-3. *International Electrotechnical Commission*, 2014.

[7] IEEE. Time-Sensitive Networking (TSN) Task Group. https://1.ieee802.org/tsn/, 2018. [Online; accessed 27-November-2018].

[8] IEEE Standard for Local and metropolitan area networks – Audio Video Bridging (AVB) Systems. *IEEE Std 802.1BA-2011*, pages 1–45, 2011.

[9] Draft Standard for Local and Metropolitan Area Networks — Virtual Bridged Local Area Networks — Amendment: Forwarding and Queuing Enhancements for Time-Sensitive Streams. *IEEE P802.1Qav/D7.0*, 2009.

[10] Christian Boiger. Class A bridge latency calculations. In *IEEE 802 November Plenary Meeting*, 2010.

[11] IEEE Standard for Local and Metropolitan Area Networks – Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks. *IEEE 802.1AS-2011*, 2011.

[12] IEEE Standard for Local and metropolitan area networks – Bridges and Bridged Networks - Amendment 25: Enhancements for Scheduled Traffic. *IEEE Std 802.1Qbv-2015*, pages 1–57, 2016.

[13] Draft Standard for Local and Metropolitan Area Networks — Bridges and Bridged Networks — Amendment: Asynchronous Traffic Shaping. *IEEE P802.1Qcr/D0.5*, 2018.

[14] J. Specht and S. Samii. Urgency-Based Scheduler for Time-Sensitive Switched Ethernet Networks. In *2016 28th Euromicro Conference on Real-Time Systems (ECRTS)*, pages 75–85, 2016.

[15] IEEE Standard for Ethernet Amendment 5: Specification and Management Parameters for Interspersing Express Traffic. *IEEE 802.3br*, 2016.

[16] IEEE Standard for Local and Metropolitan Area Networks — Bridges and Bridged Networks — Amendment 26: Frame Preemption. *IEEE 802.1Qbu*, 2016.

[17] Jean-Yves Le Boudec. A theory of traffic regulators for deterministic networks with application to interleaved regulators. pages 1–11, 2018.

[18] Ehsan Mohammadpour, Eleni Stai, Maaz Mohiuddin, and Jean-Yves Le Boudec. Latency and backlog bounds in time-sensitive networking with credit based shapers and asynchronous traffic shaping. In *30th International Teletraffic Congress (ITC 30)*, Vienna, Austria, 2018.

[19] SimPy – Discrete event simulation for Python. https://simpy.readthedocs.io/, 2018. [Online; accessed 27-November-2018].

[20] Jonathan S Turner. New directions in communications (or which way to the information age?). *IEEE Communications Magazine*, 40(5):50–57, 2002.

[21] Hui Zhang and Domenico Ferrari. Rate-controlled static-priority queueing. In *IEEE INFOCOM*, volume 93. Citeseer, 1993.

[22] Pawan Goyal and Harrick M Vin. Generalized guaranteed rate scheduling algorithms: a framework. *IEEE/ACM transactions on networking*, 5(4):561–571, 1997.

[23] Tobias Hoßfeld, Florian Metzger, and Poul E Heegaard. Traffic modeling for aggregated periodic IoT data. In *2018 21st Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, pages 1–8. IEEE, 2018.