©2021 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
Published as poster in the proceedings of 2021 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN), 2021

POSTER: Leveraging PIFO Queues for Scheduling in Time-Sensitive Networks

Christoph Gärtner[*], Amr Rizk[¶], Boris Koldehofe[‡], Rhaban Hark[*], René Guillaume[§], Ralf Kundel[*] and Ralf Steinmetz[*]
[*]Technical University of Darmstadt, Germany, {firstname.lastname}@kom.tu-darmstadt.de
[¶]University of Duisburg-Essen, Germany, amr.rizk@uni-due.de
[‡]University of Groningen, Netherlands, boris.koldehofe@rug.nl
[§]Robert Bosch GmbH, Corporate Research, rene.guillaume@de.bosch.com

***Abstract*—Time-Sensitive Networking emerged as a convergent Ethernet-based real-time networking standard for industrial applications. To support real-time, jitter-free isochronous traffic the corresponding TSN mechanism denoted Time Aware Shaper requires special hardware support. In this work, we propose a path to building TSN networks on top of programmable switches. Specifically, we show here how to leverage a data structure amenable to programmable data planes known as Push-in First-out (PIFO) queue to support TSN traffic scheduling for isochronous real-time, as well as, best effort traffic.**

***Index Terms*—TSN, IEEE802.1Qbv, PIFO, scheduling**

I. Introduction

Time-Sensitive Networking (TSN) has risen in recent years as an approach to convergent real-time networking standard with deterministic guarantees for industrial applications. Currently, TSN relies on special switching hardware which deterministic ensures latency guarantees for a handful of given scheduling mechanisms. One popular mechanism for scheduling real-time flows is the Time Aware Shaper (TAS), standardized in IEEE 802.1Qbv. It allows the programming of cyclic open and close instructions regulating queues with strict priority transmission selection at supported switch output ports. These instructions allow providing so called *scheduled traffic*, i.e. the real-time traffic class, with predefined transmission windows for jitter and loss free communication.

Push-In-First-Out (PIFO) [1] queues can be regarded as a priority queuing concept, designed for line-rate deployability in hardware. Packets can be inserted at an arbitrary position in the queue, but are always dequeued from the head. Enqueueing a packet at a certain position corresponds to the *rank* of that packet relative to the enqueued packets. This versatile concept allows expressing different types of schedulers such as priority and Least Slack Time First schedulers [1]. Current trends allow anticipating upcoming off-the-shelf switching hardware with PIFO support, e.g. as an extension to P4 data-plane programmable switches.

In this work, we propose utilizing PIFO queues to express the main functionality of the TSN Time Aware Shaper mechanism. We show how it can be used to realize non-overlapping scheduled traffic. Using a queuing model like PIFO requires the computation of a rank at the time of enqueue, and in addition, the design of an appropriate hierarchical queue structure, such that the desired scheduling algorithm can be mapped to a PIFO structure. Note that the PIFO queuing concept supports hierarchical queueing, which are drained from the root [1].

II. Scheduled Traffic using PIFO queues

The Time Aware Shaper has a port cycle time, and up to eight priority queues, that can be opened and closed subject to hardware-specific time-granularity. A frame is not transmitted out of a queue if there is not enough time available until the next gate close instruction.

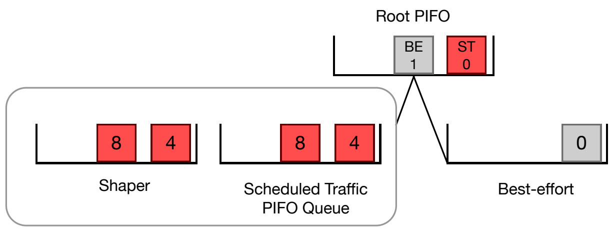


Fig. 1: Hierarchical PIFO queue structure to support scheduled traffic (ST) windows and best-effort (BE) traffic. Each queue-element is sorted by its rank. The root references PIFO queues of the leaf. The shaper is responsible for enqueuing ST references when the clock reaches the assigned rank.

We consider two classes of traffic that are supported by our approach: (a) scheduled traffic (ST), which has cyclic windows designated for pre-computed real-time flows, and (b) best-effort (BE) traffic, which does not receive service guarantees. Scheduled traffic windows are provided by scheduling algorithms such as [2]. Using a PIFO queue hierarchy as depicted in Fig. 1, we can ensure that scheduled traffic packets are transmitted in their designated time-slots. This is enabled by a secondary shaping PIFO queue, which holds back the enqueuing of references to the scheduled traffic queue into the root PIFO until the time of their designated window is reached. The scheduled time at the initial enqueue of a scheduled packet also directly gives the packet rank, i.e. its order.

A naive approach is to let the root PIFO queue prioritize all scheduled traffic and de-prioritizes all best-effort traffic by assigning a rank of 1 for best-effort and 0 for scheduled traffic at the root PIFO. However, applying this approach would introduce jitter for scheduled traffic windows. This occurs when the transmission time of best-effort traffic overlaps with the reserved window of scheduled traffic (cf. Fig. 2), and may

The documents distributed by this server have been provided by the contributing authors as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, not withstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

©2021 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
Published as poster in the proceedings of 2021 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN), 2021

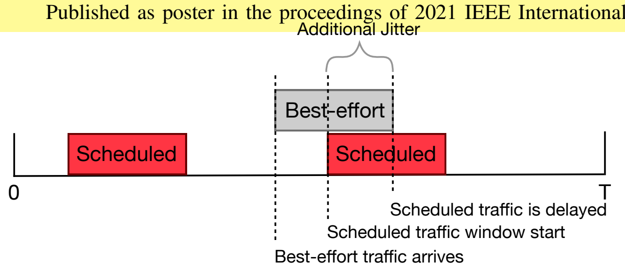

Additional Jitter
Best-effort
Scheduled
Scheduled
0
T
Scheduled traffic is delayed
Scheduled traffic window start
Best-effort traffic arrives

Fig. 2: BE packets scheduled using PIFO just before scheduled traffic windows can result in additional jitter per hop.

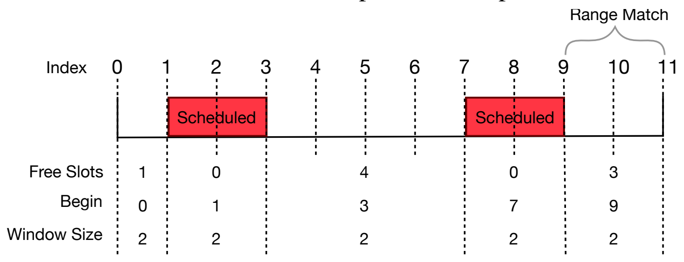
repeat if multiple packets are transmitted within the reserved window. This per-hop jitter is bounded by the transmission time of the best-effort packet MTU, e.g. $12\mu s$ at 1Gbps with 1514byte packets. In standard TAS this behavior is avoided as the gate responsible for best-effort traffic can be closed when a scheduled traffic gate is open. Furthermore, TAS switches use implicit or explicit guard-bands to ensure no BE frame overlaps onto scheduled traffic windows.

To ensure this behavior with our approach we need to provide the guard-band functionality using the PIFO concept. Since we cannot delay the BE queue like a standard TAS, BE packets need to be scheduled dynamically based on the reserved scheduled traffic windows at the corresponding port. This must be done at each hop along a network path of a stream. Note that dynamical scheduling of BE packets in the proposed approach is not trivial, since loops are not directly supported by programmable switching hardware [3]. Hence, we cannot shift the scheduled time of new BE packets until the scheduled time is cleared of the reserved window.

Our approach relies on a look-up table (LUT), as depicted in Fig. 3, to provide this behavior. By segmenting the port cycle into scheduled and non-scheduled windows, we can lookup the required information to avoid overlaps of BE packets onto scheduled traffic. For each window we keep a vector with *(i)* the number of time-units until the next reserved scheduled traffic window, *(ii)* the next scheduled traffic window-size, as well as *(iii)* the beginning time of the current window. All time-points are relative to the start of the port cycle time. The sum of a window's vector will correspond to the next *possible insertion* position relative to the beginning of a cycle. Therefore, the next `window-size` must point to a time-point, where at least one MTU-sized best-effort packet can be scheduled without overlapping onto a reserved window, i.e. reserved window gaps smaller than one MTU are ignored. The rank calculation of the best-effort shaper is sketched in Alg. 1, with `last_BE_endtime` representing the time point, at which the transmission of the last scheduled best-effort packet ends. This and the rank are the only modified state.

A LUT can be implemented within a programmable switch using match-action units. The contained tables support range matches, which in turn support the retrieval of the window-vector. In case no range matches are supported by the device, they can be realized by multiple prefix or exact matches.

The approach illustrated here is comparable to using implicit guard-bands. A best-effort packet is sent before reserved scheduled traffic windows, if the packet size permits. However, if there are gaps of less than one MTU, this approach cannot populate these gaps, due to the safety margin within the window-size variable. To support this, it would require new entries in the LUT for different possible BE packet sizes.



Free Slots	1	0	4	0	3
Begin	0	1	3	7	9
Window Size	2	2	2	2	2

Fig. 3: Look-up table encodes the number of time-units until the next reserved window, as well as the size of the next reserved window. When used with range-lookups the table must also reference the beginning of the matched range.

Algorithm 1: Rank for Best-Effort PIFO Queue Shaper

```
Data: Packet p(frameduration);
last_BE_endtime;
if last_BE_endtime < NOW then
    last_BE_endtime = NOW;
rel_pos_start = last_BE_endtime % CYCLE_TIME;
lut = LUT(rel_pos_start);
slots_available = lut.begin - rel_pos_start +
  lut.free_slots;
if slots_available < p.frameduration then
    p.rank = last_BE_endtime - rel_pos_start +
      $\sum_i$ lut.i;
else
    p.rank = last_BE_endtime;
last_BE_endtime = p.rank + p.frameduration;
```

III. Conclusion

In this work we presented a novel approach for replacing special time-sensitive networking hardware by programmable of-the-shelf switches with PIFO queues. We showed how to use PIFO queues to schedule real-time TSN traffic together with best-effort traffic. We provided an algorithm that can be directly implemented on programmable switches with a PIFO programmable traffic manager to isolate scheduled traffic from best-effort traffic using a guard-band functionality.

IV. Acknowledgment

This work has been funded in parts by the German Research Foundation (DFG) as part of projects T3, B4 within the Collaborative Research Center (CRC) 1053 – MAKI as well as the DFG project SPINE.

References

[1] A. Sivaraman, S. Subramanian, M. Alizadeh, S. Chole, S.-T. Chuang, A. Agrawal, H. Balakrishnan, T. Edsall, S. Katti, and N. McKeown, "Programmable Packet Scheduling at Line Rate," in *Proc. of 2016 ACM SIGCOMM*, 2016, pp. 44–57.
[2] S. S. Craciunas, R. S. Oliver, M. Chmelík, and W. Steiner, "Scheduling Real-Time Communication in IEEE 802.1Qbv Time Sensitive Networks," in *Proc. of ACM RTNS*, 2016, pp. 183–192.
[3] P. Bosshart, D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford, C. Schlesinger, D. Talayco, A. Vahdat, G. Varghese, and D. Walker, "P4: Programming protocol-independent packet processors," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, p. 87–95, Jul. 2014.

The documents distributed by this server have been provided by the contributing authors as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, not withstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.