Demo: Flexibility-aware Network Management of Time-Sensitive Flows

Christoph Gärtner
Technical University of Darmstadt
Germany
christoph.gaertner@kom.tu-darmstadt.de

Amr Rizk
University of Duisburg-Essen
Germany
amr.rizk@uni-due.de

Boris Koldehofe
Ilmenau University of Technology
Germany
boris.koldehofe@tu-ilmenau.de

René Guillaume
Robert Bosch GmbH
Germany
rene.guillaume@de.bosch.com

Ralf Kundel
Technical University of Darmstadt
Germany
ralf.kundel@kom.tu-darmstadt.de

Ralf Steinmetz
Technical University of Darmstadt
Germany
ralf.steinmetz@kom.tu-darmstadt.de

ABSTRACT

We investigate the application of a recently published metric for flexibility in the context of combined port queue schedules of network paths in Time-Sensitive Networks (TSN). TSN comprises a set of specifications for deterministic networking, including support for scheduled traffic with guaranteed deterministic end-to-end delays. Typically, scheduler resource allocation in TSN disregards flexibility of scheduler configurations. Essentially, the notion of flexibility of paths comprising multiple concatenated ports having each a TSN configuration is based on the number of possible embeddings, i.e., resource allocations, for a new flow of a given specification (size and delay deadline) along that path. This demonstration allows the user to define TSN schedules along network paths and, hence, illustrates the behavior and benefit of performing flexibility-aware TSN configuration.

CCS CONCEPTS

• **Networks** → **Network performance modeling**; **Network management**; *Bridges and switches*; Network servers; **Network resources allocation**; **Packet scheduling**; • **General and reference** → *Metrics*.

KEYWORDS

Time-Sensitive Networking, TSN, scheduling, flexibility, optimization, deterministic networking

ACM Reference Format:
Christoph Gärtner, Amr Rizk, Boris Koldehofe, René Guillaume, Ralf Kundel, and Ralf Steinmetz. 2023. Demo: Flexibility-aware Network Management of Time-Sensitive Flows. In *ACM SIGCOMM 2023 Conference (ACM SIGCOMM '23), September 10, 2023, New York, NY, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3603269.3610869

ACM SIGCOMM '23, September 10, 2023, New York, NY, USA

ACM ISBN 979-8-4007-0236-5/23/09.
https://doi.org/10.1145/3603269.3610869

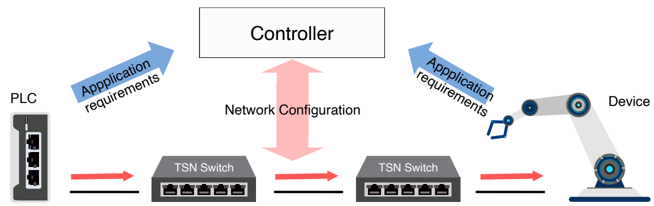

Figure 1: A controller ensures that the deterministic QoS requirements of data flows are met in a network of TSN-capable devices and switches.

1 INTRODUCTION

The Time-Sensitive Networking (TSN) extension of switched Ethernet brings new possibilities for deterministic networking to a number of challenging use cases such as critical in-vehicle communication, industrial or machine environments [1]. In scenarios involving strict traffic guarantees, the TSN Time Aware Shaper (802.1Qbv) enables a priori deterministic, jitter-free and isochronous traffic scheduling. Here, a logically centralized TSN controller can collect application requirements (cf. Figure 1) and configure the TSN network switches accordingly by encompassing a global view of all schedules at all network switch ports in addition to the QoS requirements of requested flows. To create this network configuration, TSN schedules are often computed using time-intensive constraint-based approaches, such as [3, 4].

Dynamic environments that entail frequent changes in deployed scheduled traffic flows are often disregarded in the planning phase of TSN networks. A number of approaches that aim to adapt to flow changes by featuring incremental flow admissions are given in [2, 5, 8]. However, in general, TSN systems and the correspondingly deployed schedulers do not actively consider these highly dynamic environments in the dynamic resource planning and optimization, for example, when choosing network paths for newly admitted flows. To maximize future network configuration possibilities, we propose the use of flexibility metrics as a key utility for network configuration planning before deploying individual flows. To this end, we describe and demonstrate the use of a flexibility notion for flow admission in a dynamic TSN environment.

The approach is designed for highly dynamic environments, like automated factories, where the network environment can facilitate a central controller to deploy switch configurations.

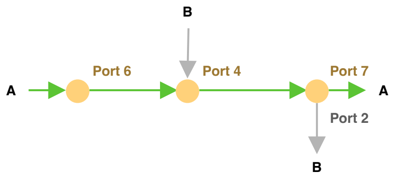

Figure 2: The selected path of interest consists of three switches with given egress ports (*A*) with another path (*B*) sharing the middle port.

2 FLEXIBILITY NOTION

To successfully introduce time dynamics to TSN configurations, particularly with deployments that handle scheduled traffic, a TSN controller requires a notion of the current scheduling context, instead of solely aiming to satisfy current application requirements. In [6, 7] a flexibility notion, named *flexcurve*, for TSN scheduled traffic is proposed, which we adopt and investigate in this demonstration. In a nutshell, given the time schedules for all switch ports along a given path, the flexcurve captures the number of possible flow embeddings for a flow to be admitted along that network path for flows of different sizes. Applying the flexcurve to a pre-selected path yields a value of possible future arrangements of new flows given their size and delay deadlines at an automatically identified bottleneck. Therefore, higher flexcurve values at the selected path grant more flexibility with the embedding, i.e. admission, of future flows. Additionally, by considering the flexcurve value on an incremental flow admission request basis, the computation of the flexcurve can also be used to provide flexibility-aware flow admissions and routing between two network end points.

3 DEMONSTRATION

We visualize the mentioned aspects of the flexcurve by isolating the view on a simple flow-path (*A*) consisting of three switches, within a larger topology. This view results in three affected egress ports and is sketched in Figure 2 from left to right. At the center hop at port 4, cross-traffic can be introduced to mix traffic of path *B* with *A*. Flows introduced with *B* leave the view at the last switch using port 2, which is not selected.

The user-facing demo-interface consists of a single-page web application, that accesses our TSN controller. The controller is responsible for scheduling all incoming flow requests and computing the necessary flexcurve values. For each flow request received we use three different strategies to embed a flow within each schedule. To achieve a very high schedule utilization across all three strategies, the number of available queues has been increased to 16. Therefore, to visualize this for the demo, the controller keeps a separate configuration state for each embedding strategy:

(1) *Flexibility Aware Strategy (approximated)*: The flexcurve is used to select the position of new flows in the schedules along a network path. To achieve faster runtimes, the selection is based on the approximate approach from [6].
(2) *Greedy Strategy*: Uses a first-fit approach to select the position of a flow to achieve fast results, but without consideration of flexibility.
(3) *Random Strategy*: Randomly chooses the position of a flow. This strategy also disregards the requested flow end-to-end delay deadline.

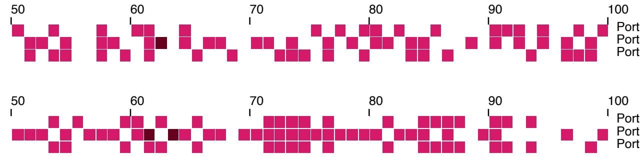

Figure 3: Example heatmap of flow assignments: The current schedule is visualized using a heatmap from 0–100%. This allows the comparison of assignments between incremental changes and strategies, e.g., from top (initialization) to bottom (240 changes).

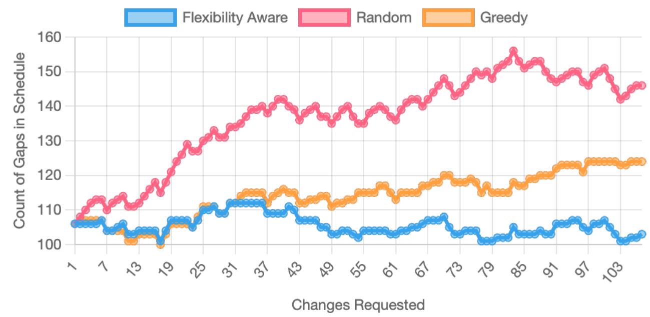

Figure 4: Visualization of the evolving fragmentation displaying the total number of gaps for each strategy after changes are introduced. Here, the flexibility-aware strategy produces the least fragmentation.

The goal of the demo is to show the behavior and benefit of flexibility-aware scheduling, and the ability to analyze flexibility regardless of scheduling strategy or algorithm in a dynamic and incremental flow scenario.

Using the web interface, schedules can be randomly initialized to continue if desired, from a non-empty state. Time is slotted and the user can start to request flow embeddings using either path *A* or *B* with random flow sizes (from 100 up to 250 slots). Flow periods and deadlines are set to 15000 slots. Furthermore, flows can also be randomly evicted from each schedule to increase the dynamicity further. Admitted flows sustain their specific slots without modification until their eviction.

After a successful flow admission, the interface displays the current aggregated flexcurve value of multiple possible flow sizes and also the current schedule fragmentation as a count of gaps (cf. Figure 4). Additionally, a heatmap of each schedule's flow allocation (cf. Figure 3) is displayed to visualize the different flow allocations between strategies. In the displayed example session, scheduling using a flexibility aware strategy resulted in less fragmentation and higher flexcurve values.

ACKNOWLEDGEMENT

This work has been funded by the German Research Foundation (DFG) as part of projects B4, T3 within the Collaborative Research Center (CRC) 1053 – MAKI. The work of A. Rizk was additionally supported by the project FlexShield.

REFERENCES

[1] Rudy Belliardi, Josef Dorr, Thomas Enzinger, Florian Essler, János Farkas, Mark Hantel, Maximilian Riegel, Marius-Petru Stanica, Guenter Steindl, Reiner Wamßer, Karl Weber, and Steven A. Zuponcic. 2018. Use Cases IEC/IEEE 60802 v1.3.

[2] Randeep Bhatia, T.V. Lakshman, Mustafa F. Özkoç, and Shivendra Panwar. 2021. FlowToss: Fast Wait-Free Scheduling of Deterministic Flows in Time Synchronized Networks. In *2021 IFIP Networking Conference (IFIP Networking)*. 1–6. https://doi.org/10.23919/IFIPNetworking52078.2021.9472838

[3] Silviu S. Craciunas, Ramon Serna Oliver, Martin Chmelík, and Wilfried Steiner. 2016. Scheduling Real-Time Communication in IEEE 802.1Qbv Time Sensitive Networks. In *Proceedings of the 24th International Conference on Real-Time Networks and Systems* (Brest, France) *(RTNS '16)*. Association for Computing Machinery, New York, NY, USA, 183–192. https://doi.org/10.1145/2997465.2997470

[4] Frank Dürr and Naresh Ganesh Nayak. 2016. No-Wait Packet Scheduling for IEEE Time-Sensitive Networks (TSN). In *Proceedings of the 24th International Conference on Real-Time Networks and Systems* (Brest, France) *(RTNS '16)*. Association for Computing Machinery, New York, NY, USA, 203–212. https://doi.org/10.1145/2997465.2997494

[5] Wen Gao, Borui Zhao, and Xu Mao. 2020. Research on Incremental Scheduling Backtracking Algorithm for Time-triggered Ethernet. In *2020 2nd International Conference on Advances in Computer Technology, Information Science and Communications (CTISC)*. 75–79. https://doi.org/10.1109/CTISC49998.2020.00019

[6] Christoph Gärtner, Amr Rizk, Boris Koldehofe, René Guillaume, Ralf Kundel, and Ralf Steinmetz. 2023. Fast incremental reconfiguration of dynamic time-sensitive networks at runtime. *Computer Networks* 224 (2023), 109606. https://doi.org/10.1016/j.comnet.2023.109606

[7] Christoph Gärtner, Amr Rizk, Boris Koldehofe, Rhaban Hark, René Guillaume, and Ralf Steinmetz. 2021. Leveraging Flexibility of Time-Sensitive Networks for dynamic Reconfigurability. In *2021 IFIP Networking Conference (IFIP Networking)*. 1–6. https://doi.org/10.23919/IFIPNetworking52078.2021.9472834

[8] Naresh Ganesh Nayak, Frank Dürr, and Kurt Rothermel. 2017. Incremental flow scheduling and routing in time-sensitive software-defined networks. *IEEE Transactions on Industrial Informatics* 14, 5 (May 2017), 2066–2075. https://doi.org/10.1109/tii.2017.2782235 Publisher: IEEE.