Enhancing Flexibility for Dynamic Time-Sensitive Network Configurations

Christoph Gärtner*, Amr Rizk¶, Boris Koldehofe‡, 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*—Today, Time-Sensitive Networks (TSN) deployments are rather static and limit to a great extend possibilities for dynamicity in real-time networking scenarios. Possible changes in the network configuration need to be accounted for already at the planning stage. Unforeseen changes mostly require complete re-configurations at the cost valuable network uptime. In order to better support dynamic reconfigurations of TSN deployments at run-time the project *DynSDN* aims to increase the online flexibility of TSN-mechanisms.**

I. Introduction

DynSDN is a transfer-project of the *DFG Sonderforschungsbereich 1053 MAKI (Multi-Mechanismen-Adaption für das künftige Internet)* and is carried out in cooperation with Robert Bosch GmbH. *DynSDN* aims to increase flexibility and adaptivity in Time-Sensitive Networking (TSN) for manufacturing scenarios by building on an increased programmability of network components.

TSN is known as a set of standards extending IEEE 802.1Q bridged Ethernet with real-time capabilities. This standardization reduces dependency on proprietary deterministic communication protocols (e.g., PROFINET, TTEthernet, and EtherCAT) and, by this, reduces vendor-dependence in hard and software due to standard interoperability. The scope of TSN mechanisms ranges from credit-based traffic, asynchronous traffic shaping, and synchronous mechanisms. For very strict real-time requirements, such as often required for *in-machine* communication, synchronous mechanisms are required.

Current deployments of strict real-time requirements with TSN limit adaptivity of highly dynamic application scenarios. As required, e.g., by dynamically connecting and disconnecting machines in a manufacturing setup that impose dynamic changes of network flows. In order to better support such dynamics, the approach researched in the context of *DynSDN* is to model and perform such dynamic changes in the form of *transitions*, i.e., a procedure describing how to perform seamless changes between the deployed TSN-mechanisms. Such transitions have been understood well and shown to support adaptivity to high dynamics in the context of network protocol configurations [1]. However, they are highly challenging to perform in a TSN, where hard real-time guarantees must be enforced in any configuration of a TSN, especially in the intermediate states of performing a transition. Therefore, in the context of *DynSDN*, we i) aim to understand and show how to increase the flexibility of existing TSN mechanisms and ii) understand to which extent dynamic changes can be enforced for such mechanisms with the help of Software-Defined Networking (SDN) components.

II. Adaptivity Requirements in TSN

To better illustrate the challenges imposed by TSN dynamics, we consider a manufacturing process in the following. In the context of the manufacturing process we introduce existing TSN mechanisms and identify sources of dynamics, i.e., application-level and flow-level dynamics, that require a high flexibility in existing TSN mechanisms.

In the following, we define a manufacturing process as a set of interacting real-time tasks. Correspondingly, each task spans a set of traffic flows required to perform the task successfully. For instance, the imposed traffic flows may be used for controlling a motor, the temperature value of a sensor, the transmission of a video stream, or monitoring the manufacturing process. In order to support timely task execution and interactions between the tasks, a flow must meet strict real-time end-to-end requirements dependent on the flow characteristics, often defining source, destination, deadline, and payload size.

The TSN Time-Aware Shaper is a prominent TSN mechanism that supports meeting strict real-time guarantees. A drawback of this mechanism is that it needs to be configured a priori, and its configuration is derived from very computation-intensive scheduling procedures without regard to future needs for network adaptivity. In the context of a manufacturing scenario, this procedure can impose that reacting to changes like integrating new devices, reaction to failures of devices, changes in the interaction patterns of tasks are expensive for the manufacturing process. The reason is that accounting for such changes will, in many cases, require the interruption of the manufacturing process until a new feasible solution can be deployed for the TSN network.

In *DynSDN*, we aim to reduce such costs by increasing the flexibility of TSN mechanisms [2], [3]. The increased flexibility can be used to perform dynamic changes to the TSN at run-time, as illustrated in Fig. 1. By dynamically collecting application requirements, the controller can perform transitions

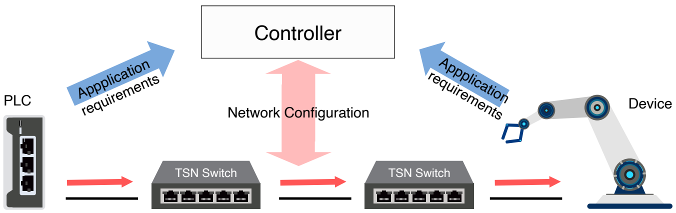

Fig. 1. A network comprised of TSN-capable devices and switches ensures traffic requirements for industrial applications.

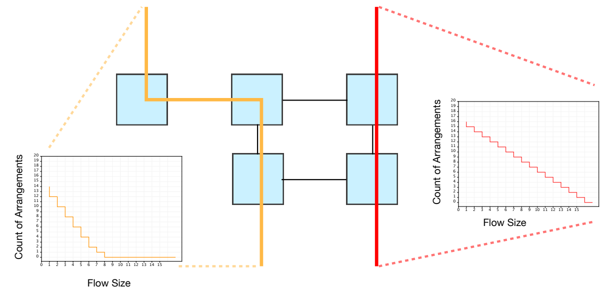

Fig. 2. Two flexcurves for disjoined paths will not impact each other. Counts of arrangement denote the number of possible flow embeddings for flows of various sizes along that path.

between deployed network configurations that allow reacting to new requirements.

In the context of the manufacturing process study in *DynSDN* currently regards two important sources for dynamic changes to the TSN. The first one relates to application-level changes, e.g., in the form of

- A new machine gets connected to the factory network.
- A machine is shutdown or removed.
- A vehicle docks and connects itself.
- A machine is changed.

Another important type of application-level changes are changes that do not impact the current task-set requirements but still require reconfiguration. For example, relocation of a source (e.g., remote Programmable Logic Controller (PLC)) due to a software update or changing network conditions requires a different routing of the required flows. The handover between the old and new sender happens at a specified time authorized by the controller when the network can be successfully adapted. Another example is the necessary re-routing of flows due to changes in network infrastructure (hardware failures/changes).

In response to application-level changes the controller needs to determine corresponding changes in the network configurations which manifest themselves in the form of flow-level changes, in particular flow admissions, flow removals, and flow modifications. *Flow admissions* enable the controller to accommodate for more interactions between the real-time tasks. It requires a careful analysis of whether new flows can be admitted without violating the real-time requirements of the entire task set. *Flow removals* allow releasing resources that can be used for further flow admissions. *Flow modifications* enable the controller to alter properties of the real-time flows which may require less or even more resources of the TSN-network. Note, that dynamic flow admission, removal, and modification can be used to control the available resources, but they can also lead to significant fragmentation of available resources.

III. Improving TSN Flexibility

To exemplary illustrate how flexibility can be enhanced, we describe in the context of flow scheduling a contribution.

The *flexibility curve* [2], in short, *flexcurve*, a notion of flexibility for scheduled traffic supported by the TSN Time-Aware Shaper. *The flexcurve provides by definition a concrete number of possible flow-embeddings along a given flow path.* As illustrated in Fig. 2 the flexcurve can be used in the course of flow admissions to reason on the impact of alternative ways (paths) admitting a flow and maximize the flexibility of the TSN mechanism to accommodate future changes.

Applications: After the initial schedule creation, the flexibility curve can be precomputed. It can then be used to provide information on the embeddability of future flows for a controller. We showed that this is much faster than by obtaining this information from calculating a new schedule. Therefore rendering resource exhausting schedule recalculations unnecessary and allowing for rapid insertions of new flows. The second class of applications is the usage as optimization criteria during the initial schedule creation. This allows the network designer either to maximize the flexibility for all distinct routes in the network or to prioritize certain future routes which are expected to be working to capacity.

IV. Conclusion

DynSDN considers adaptivity requirements we anticipate are necessary to support in future *flexible* TSN. To support this, we contribute approaches to increase flexibility. Among those contributions is the flexcurve, a notion for TSN scheduling path flexibility, supported by the Time-Aware Shaper mechanism. This enables an admission selection of flows for a central controller. In the future, we aim to increase the flexibility further by also considering different TSN mechanisms.

Acknowledgment

This work is funded by the German Research Foundation (DFG) within the Collaborative Research Center (CRC) 1053 MAKI - subproject T3 in cooperation with Robert Bosch GmbH.

References

[1] B. Alt, M. Weckesser, C. Becker, M. Hollick, S. Kar, A. Klein, R. Klose, R. Kluge, H. Koeppl, B. Koldehofe, W. R. KhudaBukhsh, M. Luthra, M. Mousavi, M. Mühlhäuser, M. Pfannemüller, A. Rizk, A. Schürr, and R. Steinmetz, "Transitions: A Protocol-Independent View of the Future Internet," *Proc. IEEE*, vol. 107, no. 4, pp. 835–846, 2019.

[2] C. Gärtner, A. Rizk, B. Koldehofe, R. Hark, R. Guillaume, and R. Steinmetz, "Leveraging flexibility of time-sensitive networks for dynamic reconfigurability," in *IFIP Networking Conference (IFIP Networking)*, 2021.

[3] C. Gärtner, A. Rizk, B. Koldehofe, R. Hark, R. Guillaume, R. Kundel, and R. Steinmetz, "Poster: Leveraging pifo queues for scheduling in time-sensitive networks," in *2021 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*, 2021, pp. 1–2.