2018 IEEE Real-Time and Embedded Technology and Applications Symposium

Demo Abstract: Slate XNS – An Online Management Tool for Deterministic TSN Networks

Slate XNS:一个在线的付费的TSN配置工具，需要硬件支持802.1AS和802.1Qbv协议

Silviu S. Craciunas Ramon Serna Oliver Wilfried Steiner
TTTech Computertechnik AG, Vienna, Austria 维也纳，奥地利
{scr,rse,wst}@tttech.com

***Abstract*—Time-Sensitive Networks are becoming the standard in real-time networking over IEEE 802 Ethernet. The set of TSN standards currently being defined by the IEEE TSN task group offer mechanisms introducing real-time behavior over specialized hardware network nodes. Out of these standards we identify those enabling the integration of deterministic real-time communication with the legacy best-effort traffic.**

In this demonstration we show the management of a simple TSN hardware setup via the Slate XNS tool-chain. We demonstrate the ease-of-use of the tools in charge of the entire network as well as the provided scheduling capabilities allowing one-click deployment of deterministic communications with properly configured IEEE 802.1Qbv and IEEE 802.1ASrev sub-standards.

I. Introduction

Stringent real-time requirements for communication are becoming increasingly important in many different application domains. Moreover, the possibility of transmitting both critical and non-critical messages over the same standard IEEE 802 Ethernet backbone is particularly interesting due to the cost-saving opportunities as well as the ease of integrating legacy systems in a real-time environment. With the help of deterministic networks, data acquisition (i.e. sensors), control loops and actuators can be distributed via a shared Ethernet network while still retaining their real-time performance. Time-Sensitive Networking (TSN), as an extension to IEEE 802 Ethernet, has emerged as the new standardized solution for the industrial control and automation domain where real-time communication behavior is a key factor. While the different sub-standards of TSN offer mechanisms for low-latency and support for real-time behavior, determinism does not come out of the box. The right combination of sub-standards, hardware support, theory and tooling is necessary in order to achieve deterministic real-time communication over deployed TSN networks. The two key elements in this context are the hardware support for the IEEE 802.1Qbv and IEEE 802.1ASrev TSN standards as well as the software needed to generate configurations for critical communication streams.

At TTTech we have enabled real-time and safety capabilities over Ethernet through TTEthernet which was successfully demonstrated in e.g. NASAs Orion project. In this demonstration we show our solution to deterministic network deployments through TSN by means of a dedicated TSN switch IP synthesized into NetLeap evaluation boards and managed via Slate XNS, a TSN scheduling software and topology builder for generating standardized TSN configurations.

II. Deterministic communication over TSN

Out of the complete set of TSN sub-standards [1], two key protocols are the foundation necessary for guaranteeing the deterministic real-time communication requirements of critical streams as well as enabling the temporal isolation between critical and non-critical communication [2]:

- IEEE 802.1ASrev defines a protocol for time-synchronization between network nodes providing a global clock reference.
- IEEE 802.1Qbv defines the *time-aware shaper* mechanism via gates that enable and disable the transmission of individual communication priorities according to a pre-computed global communication schedule (called the Gate Control List).

By combining these two sub-standards we can achieve deterministic real-time communication over Ethernet. A pre-computed communication schedule (the synthesis of which is an NP-complete problem) defines the transmission and reception times for individual frames of critical streams on each node along the routed path in the network such that there is no contention on individual links and, additionally, real-time communication requirements of streams are maintained. The global schedule can only work as expected if all nodes in the network understand the same time within a bounded precision.

While the time synchronization of nodes is achieved out-of-the box through the protocol stack defined in IEEE 802.1AS-rev, the scheduling problem is a novel challenge not covered in the standard documentation. Inherently, the time-aware shapers defined in IEEE 802.1Qbv are simple mechanisms temporally deciding the eligibility of a subset of the available priority queues based on a local schedule running on the egress function of each communication port. The necessary step to compute such a schedule enabling deterministic real-time behavior for individual streams (i.e. not for an entire priority class) is a novel problem requiring a global network view to compute a communication schedule. A well-formed schedule shall guarantee the temporal isolation of critical streams not only from non-critical streams sharing the same link but also from other critical streams with the same priority.

We have addressed this TSN-specific scheduling problem, featured in this demonstration, in [3] and [4]. Our work derived scheduling constraints defining correct TSN Gate control Lists (GCL) achieving the required determinism on the level of individual communication streams. These constraints are used to compute schedules using a Satisfiability Modulo Theories (SMT) solver. We refer the reader to [3] and [4] for an in-

978-1-5386-5295-4/18/$31.00 ©2018 IEEE
DOI 10.1109/RTAS.2018.00020

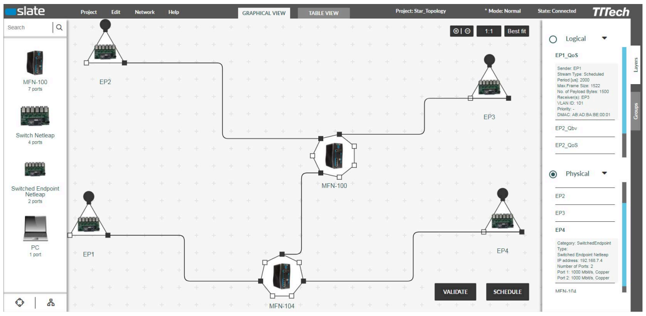

Fig. 1. Slate XNS - A Browser-based TSN Scheduling Tool

depth discussion of the TSN scheduling problem as well as our proposed solution implemented in this demonstration. In addition the the GCL schedule for each port, the devices in the network have to have a common understanding of time, otherwise the transmission points in time of individual frames will be misaligned due to the inherent clock drift. This synchronization is achieved via the protocol defined in IEEE 802.1ASrev. Hence, determinism in TSN networks can only be achieved with the support of synchronization and TSN-specific scheduling algorithms, both of them requiring hardware support in form of TSN switching engine as well as dedicated configuration tools implementing the scheduling theory from [3] and [4].

III. Slate XNS

Figure 1 shows the graphical user interface of Slate XNS[1], a commercial TSN network scheduling tool with browser-based topology modeling developed in-house at TTTech. Slate XNS makes it easy to model topologies, define critical and non-critical communication streams, create schedules, and deploy configurations. The scheduling engine implementing the theory we described in [3] and [4] is built-in to the Slate XNS software using the Z3 4.5.0 SMT solver at its core. An intuitive graphical topology modeler aids the user building the network topology, adding physical links between devices as well as communication data streams using drag-and-drop operations from a TSN device list to the topology map. The schedule can be computed and deployed to the individual nodes with just one click. New network components and streams can be added incrementally after deployment without modifying the existing schedules and configurations. Additionally, a number of user constraints, ranging from end-to-end latency requirements, start- and end-time constraints, to precedence constraints between streams, can be added to the model. Moreover, several optimization criteria can also be input by the user guiding the tool towards finding an optimal solution with respect to for e.g. end-to-end latencies of streams or the distribution of remaining unscheduled bandwidth.

The output of the tool are configuration artefacts conforming to the standard YANG models developed within the TSN task group, guaranteeing the integrability of TSN-compliant devices from third-party vendors. These artifacts are transported to the target devices using Netconf.

[1]Available at https://www.tttech.com/products/industrial/deterministic-networking/network-configuration/slate-xns/

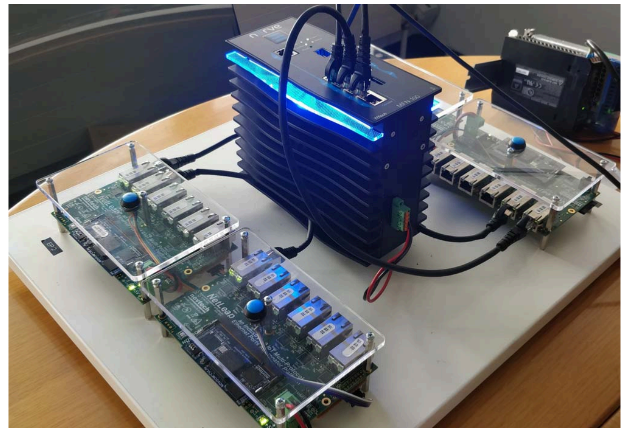

Fig. 2. The hardware demo setup

The setup is designed to support OPC UA Pub/Sub with a fully integrated publisher/subscriber stack enabling for e.g. the dynamic addition of listeners to already deployed streams. Moreover, Slate XNS schedules can be directly linked to OPC UA applications running on the end nodes.

IV. Demonstration

Figure 2 shows the hardware demo setup consisting of 4 NetLeap evaluation boards with TTTech TSN switch IP (implementing support for the IEEE 802.1Qbv Gate Control Lists and the IEEE 802.1AS time synchronization protocol) and embedded software acting as switched end-points and an MFN100 switch. The 4 switched end-points are connected to the MFN100 switch and communicate via both best effort and scheduled traffic. The setup can be reconfigured in terms of topology and/or stream definitions to show the reconfiguration capabilities of the software.

We demonstrate via this simple setup the ease-of-use of Slate XNS managing the reconfiguration of a TSN network as well as the scheduling capabilities of the tool. Additionally we show the potential of scheduled and non-scheduled traffic over TSN with the help of custom applications running on the end-nodes and generating network traffic between each other.

References

[1] Institute of Electrical and Electronics Engineers, Inc, "Time-Sensitive Networking Task Group," http://www.ieee802.org/1/pages/tsn.html, 2018, retrieved 22-Jan-2018.

[2] P. Pop, M. Lander Raagaard, S. S. Craciunas, and W. Steiner, "Design optimization of cyber-physical distributed systems using IEEE time-sensitive networks (TSN)," *IET Cyber-Physical Systems: Theory and Applications*, vol. 1, no. 1, pp. 86–94, 2016.

[3] S. S. Craciunas, R. Serna Oliver, M. Chmelik, and W. Steiner, "Scheduling Real-Time Communication in IEEE 802.1Qbv Time Sensitive Networks," in *Proc. RTNS*. ACM, 2016.

[4] R. Serna Oliver, S. S. Craciunas, and W. Steiner, "IEEE 802.1Qbv gate control list synthesis using array theory encoding," in *Proc. Real-Time and Embedded Technology and Applications Symposium (RTAS)*. IEEE, 2018, to appear.