A Simulation Model for Investigating Clock Synchronization Issues in Time-Sensitive Networks

P. Danielis[1], H. Parzyjegla[1], Y. Rashid[1], E. Schweissguth[2], G. Mühl[1], D. Timmermann[2]
[1]*Institute of Computer Science, University of Rostock*, Rostock, Germany
{peter.danielis;helge.parzyjegla;gero.muehl}@uni-rostock.de
[2]*Institute of Applied Microelectronics and Computer Engineering, University of Rostock*, Rostock, Germany
{eike.schweissguth;dirk.timmermann}@uni-rostock.de

***Abstract*—The IEEE 802.1 Time-Sensitive Networking (TSN) task force has set guidelines for IEEE 802.3 networks allowing deterministic realtime communication over Ethernet. To execute deterministic time-triggered scheduling operations, the network nodes need to be synchronized. One application of time synchronization is distributed data acquisition for the Wendelstein 7-X nuclear fusion experiment. Both the system control and the scientific evaluation of the experiments need measurement timestamps with accuracies in the nanosecond range. The TSN standard IEEE 802.1AS has specified the generalized Precision Time Protocol (gPTP) which executes the synchronization process. However, there are some issues that are not addressed in the standard such as the impact of clock skew and drift of network nodes on the number of resynchronizations needed to maintain the required synchronization accuracy. This paper introduces an OMNeT++ simulation model, which can be used to investigate clock synchronization issues in time-sensitive networks.**

I. INTRODUCTION

Realtime (RT) communication for process control and manufacturing systems is mandatory for modern industrial automation. The IEEE 802.1 Time-Sensitive Networking (TSN) task group defines a set of standards for time-sensitive data transmission over Ethernet networks. TSN makes it conceivable to transport time-critical data over a bridged Ethernet network with zero packet loss [1]. To enable RT communication, the nodes of time-sensitive networks need to be synchronized. There are many applications for time synchronization in the (Industrial) Internet of Things [2]. Production robots working together in a production line must be precisely synchronized in a smart factory. Accurate time synchronization is also essential when multiple motors move a mechanical load or, more generally, when drives work together. In a smart power plant or in the field of energy supply, time synchronization is crucial both for analyzing blackouts and for controlling the stability of the power grid. The TSN standard IEEE 802.1AS defines a time synchronization protocol called generalized Precision Time Protocol (gPTP), which is an extension of PTP [3]. It specifies the procedure of how to synchronize the time in a distributed time-aware system (TAS) for time-sensitive applications in 802.1 bridged local area networks (LANs). These time-aware systems set up a common reference time within the LAN. An 802.1 bridged network is built up by two types of systems, i.e., time-aware bridges and time-aware end stations. A time-aware end station can also be selected as a grandmaster (GM). The GM provides the timing information to all time-aware systems connected in the network. All other time-aware systems set their local clock according to the GM.

There are certain factors such as temperature, aging, crystal frequency, and the manufacturing process of the clock that cause a continuous clock deviation. This clock deviation can have a significant impact on the scheduling process of time-sensitive data such as missing the deadlines for data delivery, which may further lead to drastic consequences for the result of the RT operation. However, the 802.1AS standard does not address the question of how often network nodes need to be resynchronized to compensate the clock deviation in order to meet given precision constraints.

The main contribution of this paper, described in Section II, is the integration of a simulation model based on [4], which implements IEEE 802.1AS, into the NeSTiNg project, which is an OMNeT++ simulation model for time-sensitive networks [5]. To simulate deviating clocks on network devices, the simulation model uses a clock model that simulates clock drift. The resynchronization intervals that are needed to be able to adhere to given schedules when clocks deviate continuously, can be analyzed with the developed model. The results analysis given in Section III is based on data obtained from the simulation model. The paper concludes in Section IV.

II. SIMULATION MODEL DESIGN AND IMPLEMENTATION

There are three OMNeT++ frameworks needed for designing the simulation model. Firstly, INET serves as the basis for Ethernet simulation models. Secondly, NeSTiNg builds upon this and is used to leverage TSN end-point and switch modules with IEEE 802.1Qbv time-aware shaper functionality to schedule time-triggered traffic [5]. The so-called *legacyClock* module, developed by the NeSTiNg community, is used as clock model since it includes the ability to simulate clock drift. Thirdly, the gPTP functionality from [4], to be used by the TSN modules, is integrated into NeSTiNg. The NeSTiNg framework was designed to work with INET version 4.1.2 and OMNeT++ version 5.5.1. However, the gPTP simulation model was originally developed for INET version 3.6.3 and

We would like to thank the German Research Foundation (DFG) (project ”Real-Time Publish/Subscribe for the Industrial Internet of Things”, GZ: TI 254) for their financial support.

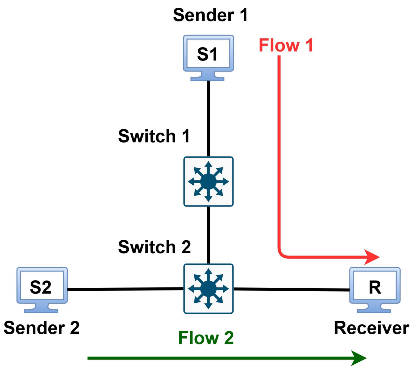

Fig. 1. Simulation setup.

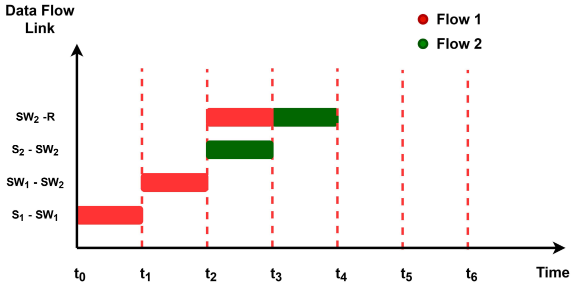

Fig. 2. Scheduling in ideal scenario.

OMNeT++ version 5.3. Hence, it was first ported to INET version 4.1.2 and then integrated into NeSTiNg.

Integration of gPTP into NeSTiNg: There are two types of time-aware systems in the gPTP network domain: time-aware end stations and time-aware bridges. NeSTiNg's end-station and bridge modules have therefore been modified to integrate the gPTP clock synchronization model into NeSTiNg.

In the end stations, the *EthernetInterface* module is replaced by the *EthernetInterfaceGPTP* module, which adds gPTP functionality to the end stations. In order to forward time synchronization information to the master ports of the time-aware systems, *EthernetinterfaceGPTP* modules must communicate with one another. A simple module called *tableGPTP*, designed by the gPTP clock synchronization model, is also implemented in order to accomplish this intercommunication. In addition, NeSTiNg's default *IClock* module is replaced with the *legacyClock* module to simulate clock drift.

The *EthernetInterface* module in the NeSTiNg bridge is replaced by the *EthernetinterfaceGPTP* module and the module *TableGPTP* is added to it. The default clock is also substituted with the *legacyClock* module. A gPTP-capable bridge can contain multiple *EthernetInterfaceGPTP* modules, but only one *tableGPTP* module.

III. SIMULATION AND EVALUATION

This section presents first simulation results obtained from a simple network in OMNeT++.

Simulation Setup: As proof of concept, we have set up a network consisting of three end stations and two switches as apparent from Figure 1. There are two types of packet flows in our simulation. In flow 1, the sender 1 (S1) sends data packets to the receiver (R) via switch 1 (SW1) and switch 2 (SW2). In flow2, the sender 2 (S2) sends data to the receiver R via switch 2. In an ideal situation, when the network devices are synchronized, the network switches and end stations precisely follow the scheduling as apparent from Figure 2.

Evaluation: The network depicted in Figure 1 was used to generate simulation results with and without clock drift and to test the effect of periodic resynchronization to correct the drift. S1 acts as GM with an ideal clock while S2 is simulated with and without clock drift. There is no clock drift in the switches. Data packets are sent every 100μs and the network nodes are resynchronized every 130μs and every 160μs, respectively. In the scenario without clock drift on S2, the scheduling works as expected (cf. Figure 2). As expected, we observe that for increasing magnitudes of drift on S2 (50, 100, 500, 5000μs/s), S2's actual sending behavior deviates more and more from the scheduled one. However, if periodic resychronization is performed S2's deviation of the sending behavior from its scheduled one is bound, which demonstrates that our simulation model works according to its specifications. As expected, our findings also show that if the clock drift is high, the resynchronization interval should be short, however, depending on the required clock accuracy. If the clock drift is smaller, the resynchronization interval can be longer to reduce synchronization-traffic overhead.

IV. CONCLUSION

This paper introduces an OMNeT++ simulation model to investigate the impact of clock synchronization issues on the end-to-end latency for scheduled traffic. A simulation model that implements clock synchronization with gPTP is integrated into the NeSTiNg project that is able to simulate time-sensitive networks. As proof of concept, a resynchronization mechanism is implemented to compensate the clock drift of network components. Simulations are performed with different clock drift values and resynchronization periods. The results show that the simulation model works as expected and show the impact of the length of the resynchronization interval on the clock accuracy in the presence of clock drift.

REFERENCES

[1] J. Farkas, L. L. Bello, and C. Gunther, "Time-sensitive networking standards," *IEEE Communications Standards Magazine*, vol. 2, no. 2, pp. 20–21, 2018.

[2] H. Puttnies, P. Danielis, A. R. Sharif, and D. Timmermann, "Estimators for time synchronization-Survey, analysis, and outlook," *IoT*, vol. 1, no. 2, pp. 398–435, 2020.

[3] IEEE Standards Association, "IEEE Standard for Local and Metropolitan Area Networks-Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks," *IEEE Std*, vol. 802, 2011.

[4] H. Puttnies, P. Danielis, E. Janchivnyambuu, and D. Timmermann, "A Simulation Model of IEEE 802.1 AS gPTP for Clock Synchronization in OMNeT++," in *OMNeT++ Summit*, 2018, pp. 63–72.

[5] J. Falk, D. Hellmanns, B. Carabelli, N. Nayak, F. Dürr, S. Kehrer, and K. Rothermel, "NeSTiNg: Simulating IEEE time-sensitive networking (TSN) in OMNeT++," in *2019 International Conference on Networked Systems (NetSys)*. IEEE, 2019, pp. 1–8.