

A Data-driven Simulation Framework for Logical 5G-TSN Bridges

Lucas Haug, Frank Dürr, Simon Egger

Institute of Parallel and Distributed Systems

University of Stuttgart

Stuttgart, Germany

{lucas.haug, frank.duerr, simon.egger}@ipvs.uni-stuttgart.de

Elena Mostovaya

Institute of Parallel and Distributed Systems

University of Stuttgart

Stuttgart, Germany

st169601@stud.uni-stuttgart.de

James Gross, Gourav Prateek Sharma

School of Electrical Engineering & Computer Science

Royal Institute of Technology (KTH)

Stockholm, Sweden

{jamesgr, gpsharma}@kth.se

Joachim Sachs

Ericsson Research

Bremen, Germany

joachim.sachs@ericsson.com

Abstract

The integration of Time-Sensitive Networking (TSN) into 5G systems is essential to support emerging real-time mobile applications. As part of this effort, 3GPP has introduced the concept of logical 5G TSN bridges, which implement the same functionality as wired TSN bridges, such as the Time-aware Shaper (IEEE 802.1Qbv). The port-to-port delay characteristics of logical 5G bridges, however, are fundamentally different to those of wired TSN bridges. In this paper, we present an open-source OMNeT++/INET-based simulation framework with data plane models to simulate the characteristic packet delay of logical 5G bridges including novel 6G concepts such as Packet Delay Correction (PDC) as well as control plane models to simulate the dynamic behavior of converged 5G/TSN networks.

I. INTRODUCTION

The timely transmission of data is essential for many safety-critical cyber-physical systems operating over real-time communication networks. Particularly, applications involving mobility rely on wireless connectivity with stringent real-time requirements. Prominent examples include the coordinated movement of automated guided vehicles within factory environments and drones employed in smart farming to safeguard wildlife during harvesting operations (refer to [1] for additional use cases).

The need for bounded end-to-end packet delay (PD) and packet delay variation (PDV) in real-time communication networks has led to extensive standardization efforts. Particularly, the IEEE *Time-Sensitive Networking (TSN)* Task Group has specified a set of standards. These standards for example define traffic shaping mechanism which allow to shape and schedule traffic on bridges along the data path through the network. One such shaper is the Time-aware Shaper (TAS) which employs a transmission gates controlled by a time-driven Gate Control List (GCL). This allows to specify time slots when frames of egress queues with different priorities are eligible for transmission.

To support time-sensitive applications in mobile environments, the 3GPP has introduced TSN support in 5G networks by defining a logical TSN bridge architecture (cf. Figure 1). Such a logical 5G TSN bridge provides the same functionality as a traditional wired TSN bridge. It contains so-called TSN Translators (TTs). One TT connecting the network side (NW-TT) and multiple device-side TTs (DS-TTs) each connected to a mobile User Equipment (UE). The TTs inside the logical bridge act like “half-bridges” implementing ingress TSN functionality such as Per Stream Filtering and Policing (PSFP) on the ingress TT and egress TSN functionality such as TAS on the egress TT. Most importantly, the wireless link between a UE and the base station (gNB) is hidden inside the bridge, i.e., internally connecting two ports of the logical bridge. This significantly affects the non-functional behavior of the logical bridge, particularly its port-to-port delay (also referred to as bridge delay in the standards).

In a previous work, we have measured the port-to-port delay of wireless logical bridges using 5G technology (commodity of the shelf 5G equipment and Open-RAN-based hardware) [2]. Figure 2 shows fundamental differences

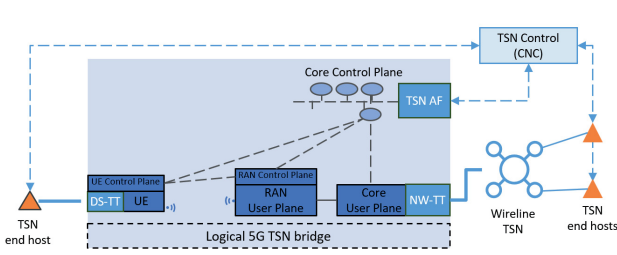
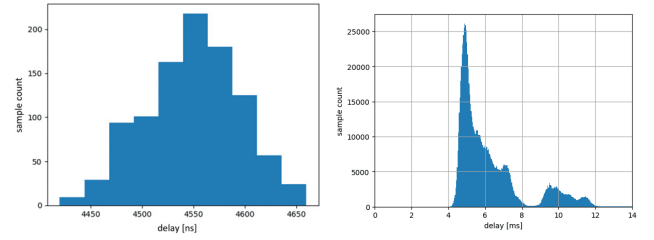


Fig. 1: Logical 5G TSN bridge.



(a) Wired bridge (b) Logical 5G bridge

Fig. 2: Port-to-port delay distributions.

of this port-to-port delay compared to wired TSN bridges: First, the port-to-port delay of a logical bridge is orders of magnitude greater (milliseconds compared to microseconds). Second, the port-to-port delay of a logical bridge spreads across an interval of multiple milliseconds (instead of a few hundred nanoseconds for wired TSN bridges) and follows a heavy-tailed bi-modal distribution. Moreover, for a heavy-tailed distribution, the probability of large values does not decrease exponentially, in contrast to the exponential latency distribution of the wired TSN bridge. Furthermore, mobile networks are typically subject to dynamic behavior. This for example includes devices joining and leaving the network as well as changes to the delay distributions due to fast and slow fading effects.

These fundamental differences regarding the port-to-port delay of logical bridges as well as the dynamic behavior of mobile networks raise multiple questions in the context of time-driven scheduling approaches, for example: (1) How does the stochastic port-to-port delay impact time-driven schedules which typically rely on precise timing; (2) How should a scheduler react on dynamic changes in the network. While there are already simulation frameworks implementing 5G models (e.g. Simu5G [3]) and frameworks implementing TSN models (e.g. INET [4]), none of these implement models of both domains. So in order to be able to answer these and further questions, we developed the first open-source simulation framework, which enables the evaluation and validation of converged 5G/TSN networks including logical bridges. Our GitHub repository of the *6GDetCom Simulator* [5] contains all models presented in this work, including a detailed documentation.

The remainder of this work describes this simulation framework in detail. In Section II, we present the data plane models including the simulation of characteristic packet delays of logical bridges. In Section III, we present our control plane models together with our scheduler integration. Finally, we conclude this paper in Section IV.

II. DATA PLANE SIMULATION MODELS

In order to allow for the simulation of converged 5G/TSN networks, there are two basic design options: (1) Develop a sophisticated model of 5G system which includes everything introducing delay along the data path (e.g., HARQ and changes to the physical environment). (2) Utilize measurements from real-world 5G systems and delay packets passing the logical bridge according to these measurements.

Due to the high complexity of implementing a realistic 5G model, our simulation framework is based on option (2). To this end, we implement a so-called DetCom node for the OMNeT++/INET framework which acts like a logical bridge. Specifically, this DetCom node consists of one NW-TT and multiple DS-TT which all inherit the TSN functionality of a typical INET TSN bridge (e.g. the TAS). Furthermore, it treats the 5G internals as a “black box” and instead utilizes our latency histograms from real 5G measurements [6].

To this end, the DetCom node is extended by a so-called Delayer component, which is located close to the ingress/egress ports of the DS-TTs for downlink/uplink delays respectively. The Delayer is either configured using latency histograms or delay traces from measurements. Different configurations can be applied in upstream and downstream direction for each DS-TT separately. Note that a packet from one UE to another UE passes the wireless link to/from the gNB two times (i.e. it is delayed using the uplink histogram of the first DS-TT and the downlink histogram of the second DS-TT), whereas a packet from/to the NW-TT only traverses the internal wireless link once. The configuration of the delayer module is flexible and for example also allows to specify delays using a random-walk process, mathematical expressions or replaying delay traces to simulate delay correlations between different links within the DetCom node. However, even with delay traces, one drawback remains: other parameters

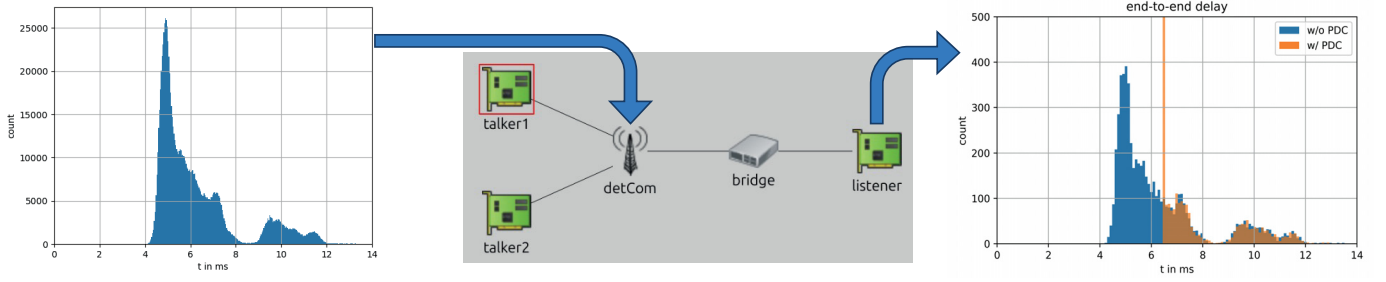


Fig. 3: DetCom node applying delay to packets with one stream using PDC.

in the simulation, such as packet size and sending rate, do not influence the resulting DetCom delays, as these are solely defined by the provided delay definition.

Additionally to the characteristic packet delay of a logical bridge, our simulator implements novel 6G concepts such as Time Synchronization in converged 6G/TSN networks and Packet Delay Correction (PDC). PDC is described in detail in [7]. In summary, PDC allows to reduce the PDV of a packet by utilizing the hold and forward buffering mechanism at the outgoing TT. More specifically, the timestamp-based implementation of PDC allows to specify a desired minimum packet delay pdv . For every frame traveling through the DetCom node, a residence time t_{res} is calculated. The outgoing TT then holds the frame for $t_{hold} = pdv - t_{res}$ in case pdv is not yet exceeded.

Figure 3 shows an example simulation with the input distribution provided on the left and the result end-to-end delay distribution of two streams on the right. It shows, that the simulated delay of packets passing through a DetCom node without PDC (blue) closely follows the original input histogram from the real network. Another stream in the same network is configured to use PDC such that 70 % of the frames have a shorter delay than the configured PDC value ($pdv \approx 6.5$ ms). The output distribution (orange) shows an accumulation in one bin at the pdv value and afterwards also follows the input distribution.

III. CONTROL PLANE SIMULATION MODELS

To comprehensively validate scheduled time-critical traffic in converged 5G/TSN networks, it is essential to analyze not only static data plane behavior but also the impact of dynamic changes within the network. Such dynamic behavior for example includes: **(1) Variable Stream Sets:** Changes to the stream sets occur, when devices join or leave the network, or when existing devices start or stop applications. These scenarios necessitate updating the currently deployed schedule, as new streams must be integrated or existing ones removed. Moreover, devices may alter their communication requirements (e.g., adjusting its sending rate), thereby requiring further schedule recalculations. **(2) Changing Packet Delay Distributions:** Wireless channel conditions are subject to temporal variations, primarily driven by slow and fast fading effects. Slow fading typically arises from large-scale mobility, such as users moving within buildings, causing gradual changes in signal strength and consequently influencing delay distributions. In contrast, fast fading, resulting from multipath propagation or minor positional adjustments, causes rapid and short-term fluctuations in delay. Both fading types necessitate adaptive scheduling strategies to maintain deterministic network performance. **(3) Variable Topology:** Mobility of devices or network failures resulting in broken links may alter the network topology. Such topology changes also require recalculations of the schedule.

Our simulation framework provides a direct, bi-directional scheduler interface with support for all types of dynamic behavior as described above. The dynamics of our scenario is defined using INET's ScenarioManager. Firstly, this allows modifying the characteristic packet delay of the Ds-TTs described in the previous section. Secondly, we provide a new app module, which can request changes to its parameters, such as the packet size, sending rate, and QoS requirements.

Upon detecting any changes in the simulation, our monitor module calls a user-defined scheduler program (e.g. a python script) providing it with the current topology, streamset and delay distributions. This especially means that our scheduler interface can be implemented with every scheduler as long as it adheres to the input and output file format. The simulation pauses during the runtime of the scheduler and afterwards applies the calculated schedule to the GCLs of the TSN bridges at the commit time provided by the scheduler.

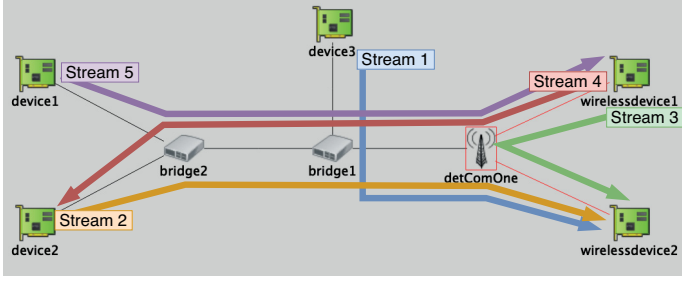


Fig. 4: Example dynamic scenario with five streams.

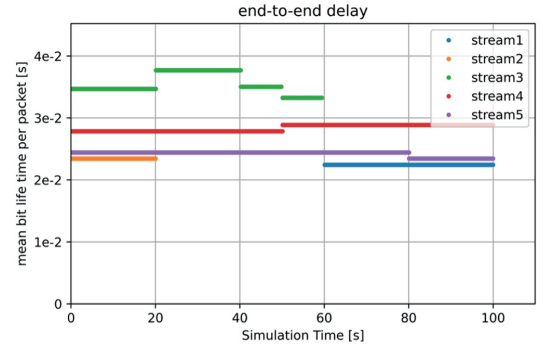


Fig. 5: Resulting end-to-end delay of dynamic scenario.

A description of the scheduler is out-of-scope for this work. However, we provide a showcase using an example scheduler. The topology and streamset of this scenario is provided in Fig. 4. The resulting end-to-end delay is shown in Fig. 5 shows the following behavior for an example dynamic scenario: (1) At $t = 20$ s: stream2 requests to stop its stream. After the execution of the scheduler, the stream is stopped, which is visible in the results by the ending orange line. At the same time the scheduler re-schedules stream3 resulting in a change of its end-to-end delay. (2) At $t = 40$ s: stream3 wants to increase the production interval and packet size. This again leads to a re-scheduling of stream3 resulting in the change of its end-to-end delay. (3) At $t = 50$ s: the downlink histogram of dstt1 improves (shifting 1 ms to the left), while the uplink histogram of dstt0 degrades (shifting 1 ms to the right) resulting in rescheduling of stream3 and stream4. (4) At $t = 60$ s: stream1 requests to start sending new data, while at the same time stream3 immediately stops sending data. This becomes visible in the results by the ending green line and the newly starting blue line. (5) At $t = 80$ s: the downlink histogram of dstt0 improves (shifting 1 ms to the right), resulting in a reduced end-to-end delay for stream5. This shows that our framework is capable of simulating various types of dynamic behavior typically occurring in converged 5G/TSN networks.

IV. CONCLUSION AND FUTURE WORK

In this work, we presented a data-driven simulation framework for logical 5G TSN bridges including data-plane models to simulate the characteristic packet delay of logical bridges as well as novel 6G concepts. Furthermore, we presented the support for dynamic behavior and our direct scheduler interface.

Next steps include the application of our simulations framework to validate our algorithms for calculating wireless-friendly TAS schedules that are robust to high packet delay variation.

ACKNOWLEDGMENTS

This work was supported by the European Union’s Horizon Europe project DETERMINISTIC6G under grant agreement No. 101096504.

REFERENCES

- [1] D. Patel, E. M. de Oca, H. N. Nguyen, J. Costa-Requena, J. Gross, G. P. Sharma, L. Grosjean, J. Sachs, J. Harmatos, O. Höftberger, F. Profelt, D. Puffer, D. Houatra, F. D. Simio, G. Bigoni, F. Giovacchini, and G. Bag, “DETERMINISTIC6G use cases and architecture principles – Deliverable D1.1 of DETERMINISTIC6G project,” <https://deterministic6g.eu/images/deliverables/DETERMINISTIC6G-D1.1-v1.0.pdf>, Jun. 2021.
- [2] S. Mostafavi, M. Tillner, G. P. Sharma, and J. Gross, “EDAF: An end-to-end delay analytics framework for 5G-and-beyond networks,” in *Proceedings of IEEE INFOCOM 2024 – 11th International Workshop on Computer and Networking Experimental Research using Testbeds (CNERT 2024)*, Vancouver, Canada, May 2024, DOI: 10.1109/INFOCOMWKSHPS61880.2024.10620853.
- [3] G. Nardini, D. Sabella, G. Stea, P. Thakkar, and A. Virdis, “Simu5G – an OMNeT++ library for end-to-end performance evaluation of 5G networks,” *IEEE Access*, vol. 8, pp. 181 176–181 191, 2020.
- [4] L. Mészáros, A. Varga, and M. Kirsche, *INET Framework*. Cham: Springer International Publishing, 2019, pp. 55–106. [Online]. Available: https://doi.org/10.1007/978-3-030-12842-5_2
- [5] “6GDetCom simulator,” https://github.com/DETERMINISTIC6G/6GDetCom_Simulator, last accessed March 2025.
- [6] “DETERMINISTIC6G measurement data,” https://github.com/DETERMINISTIC6G/deterministic6g_data, last accessed March 2025.
- [7] J. Sachs, A. Sahbafard, J. Gross, G. P. Sharma, B. Varga, and M. D. A. Jardim, “First report on 6G centric enablers – Deliverable 2.1 of DETERMINISTIC6G project,” <https://deterministic6g.eu/images/deliverables/DETERMINISTIC6G-D2.1-v2.0.pdf>, Dec. 2023.