A Deterministic Time-Sensitive Network (TSN) Simulation Model

based on OMNET++

Abdullah Al Azad, *C213106*, *Section: 7AM*, Dept. of CSE, IIUC, Chittagong, Bangladesh

Abdur Rahman, *C2130031*, *Section: 7AM*, Dept. of CSE, IIUC, Chittagong, Bangladesh

Al Fahad, *C213001*, *Section: 7AM*, Dept. of CSE, IIUC, Chittagong, Bangladesh

Md Ariful Islam Emon, *C213026*, *Section: 7AM*, Dept. of CSE, IIUC, Chittagong, Bangladesh

Md Ataul Islam, *C213107*, *Section: 7AM*, Dept. of CSE, IIUC, Chittagong, Bangladesh

***Abstract*—Time-Sensitive Networking (TSN) provides deterministic communication essential for the effective operation of industrial automation and control systems. This project, undertaken by the Network Navigators, focuses on the development of a TSN simulation model utilizing the OMNeT++ simulation environment, specifically leveraging the INET and CoRE4INET frameworks. The model features a TSN-enabled switch that employs Gate Control Lists (GCLs) for efficient traffic scheduling, thereby guaranteeing deterministic end-to-end latency. Our key contributions was to implement the GCL-driven switch model, the design of a representative network topology, and the formulation of GCL calculations based on various network parameters with considering the real situation. This simulation model serves as a platform for the evaluation of TSN scheduling algorithms and the synthesis of GCL methodologies.**

***Index Terms*—Time-sensitive Networking (TSN), OMNET++, Simulation Model, Deterministic, Real-Time, Schedule Traffic**

I. Introduction

Ethernet is the most widely utilized general-purpose communication protocol, renowned for its excellent bandwidth, scalability, and compatibility. However, meeting the real-time deterministic requirements of industrial control systems poses significant challenges. While several Ethernet extensions have provided deterministic solutions (e.g., PROFINET, EthernetCAT, and TTEthernet), they often lack mutual compatibility or do not integrate seamlessly with standard Ethernet networks and devices.

Time-Sensitive Networking (TSN) represents a significant advancement in communication technology, particularly in the context of industrial automation and control systems where deterministic communication is paramount. The ability to guarantee predictable data delivery and low latency is essential for the seamless operation of time-sensitive applications, which often require stringent timing constraints. This research focuses on the development of a TSN simulation model utilizing the OMNeT++ framework, enhanced by the INET and CoRE4INET libraries. The model incorporates a TSN-capable switch that employs Gate Control Lists (GCLs) for effective traffic scheduling, thereby ensuring reliable end-to-end latency. By creating a GCL-driven switch model and a representative network topology, this study provides a comprehensive platform for evaluating various TSN scheduling algorithms and GCL techniques. The implications of this research extend beyond the immediate capabilities of TSN, as it identifies potential areas for enhancement and encourages further investigation into adaptive scheduling mechanisms. As TSN continues to evolve from traditional Ethernet technology, its protocols for time synchronization and traffic management are increasingly recognized for their effectiveness in real-time environments, thereby laying the groundwork for future innovations in complex industrial networks. This introduction sets the stage for a deeper exploration of the methodologies and findings presented in the subsequent sections of this research.

The TSN comprises a set of IEEE 802 Ethernet sub-standards, including:

1) IEEE 802.1ASrev: Time synchronization.
2) IEEE 802.1Qbv: Traffic shaping.

These standards collectively ensure a deterministic service that guarantees packet transmission with bounded low latency, minimal packet delay variation, and low packet loss, thereby benefiting industrial control and automation.

To enhance bandwidth utilization, time-critical control data must be transmitted in a time-sensitive manner alongside other traffic classes within the same network. The challenge lies in scheduling traffic to ensure the real-time transmission of time-critical data. While prioritization was initially proposed as a solution, it proved insufficient for addressing the real-time requirements of critical traffic. The IEEE 802.1Qbv standard introduced a method termed "Enhancements for Scheduled Traffic," allowing switches to utilize GCLs to control the gate state (open or closed) of corresponding queues in the output port. This enables time-triggered (TT) traffic and best-effort (BE) traffic to be scheduled using a globally synchronized clock, ensuring real-time, predictable, and deterministic network transmission.

Identify applicable funding agency here. If none, delete this.

Despite the advancements, few studies have developed TSN simulation models. This paper presents a TSN simulation model based on OMNET++ that schedules network traffic in a time-based manner using GCLs. The contributions of this work include the development of a TSN-enabled switch capable of reading GCLs and operating the gate states of corresponding queues, the construction of a network topology to verify the switch model, and the calculation of GCLs based on network topology and host parameters. The model can also evaluate the feasibility of various traffic scheduling algorithms and GCL synthesis.

II. Overview of OMNeT++ Simulation Tool

OMNET++ is an open-source, discrete event simulator featuring an extensible, modular, component-based C++ simulation library and framework, primarily utilized for building network simulators. The advantages of OMNET++ include its open-source nature, ease of operation, and extensibility. OMNET++ provides numerous extended frameworks that support various Ethernet protocols and features, including the IEEE 802 standards. This study employs two frameworks: the INET framework, which is an open-source model library for Internet stack, wired, and wireless link-layer protocols, and the CoRE4INET framework, which extends the INET framework to allow event-based simulation of real-time Ethernet.

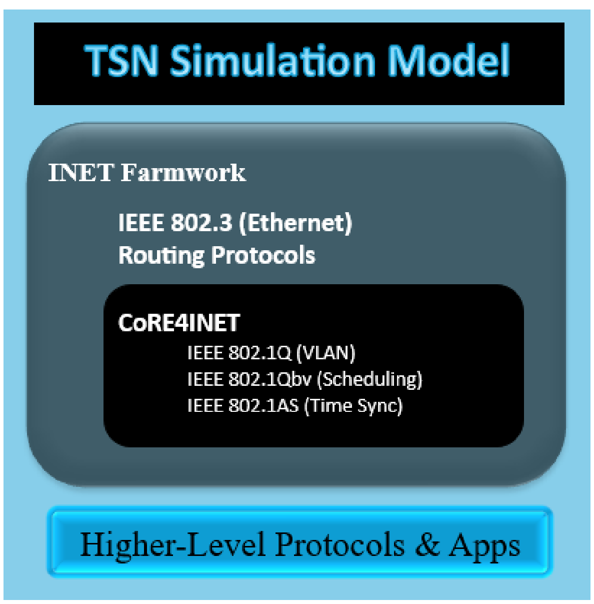

Fig. 1. Simulation Record Diagram

III. Development of the TSN Simulation Model

The development of the TSN simulation model involves two primary steps: the creation of sub-models (the TSN-enabled switch and network topology) and the synthesis of GCLs based on traffic parameters and network topology.

A. *Switch Model*

Using the extension to the IEEE 802.1Q switch provided by the CoRE4INET framework, we developed our switch model with the aid of IEEE 802.1Qbv, which allows different queues to select traffic based on their priorities and facilitates the transmission of specific traffic classes at designated times. The switch model incorporates functions for reading GCLs and managing traffic transmission.

*1) **Read Gate Control Lists**:* The switch reads parameters such as "Current Time," "Gate State," and "Time Interval." The "Current Time" indicates the moment the switch reads GCLs. When the earliest traffic reaches the switch, it triggers the reading of GCLs, updating the "Current Time" by accumulating previous values with the "Time Interval."

*2) **Traffic Transmission**:* The transmission function operates in the MAC layer of the Ethernet switch and consists of three sub-functions: "Store Traffic," "Control Gate," and "Forward." The "Store Traffic" function stores different types of traffic in eight queues based on their priorities. The "Control Gate" function checks the current gate state and updates it if necessary. The "Forward" function manages the forwarding of traffic based on the gate state and the time interval.

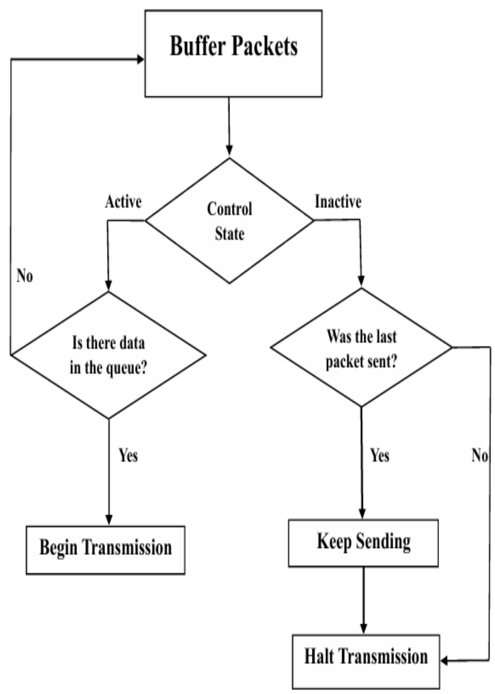

Fig. 2. Flow chart of gate operation during traffic transmission.

B. *Network Topology Architecture*

The TSN guarantees real-time deterministic transmission of time-triggered traffic in a mixed traffic scenario. To explore the worst-case conditions, we constructed a TSN topology involving mixed traffic, including a TSN talker, a best-effort sender, and a traffic generator. The nodes are interconnected with 1-Gbit/s full duplex physical links, allowing bidirectional communication. The characteristics of the traffic sent by the TSN talker, the best-effort sender, and the traffic generator are defined in a traffic parameters table.

C. *GCL Calculation*

GCLs are synthesized based on network topology and traffic parameters to ensure that only specified traffic classes can access certain channels at specific times. The GCLs create three transmission modes: Protected Window, Unprotected Window, and Guard Band. The Protected Window allows only time-triggered traffic to be transmitted, while the Unprotected Window permits the transmission of best-effort and interference traffic. The Guard Band ensures that the link is idle before the Protected Window opens, preventing any interference with time-triggered traffic.

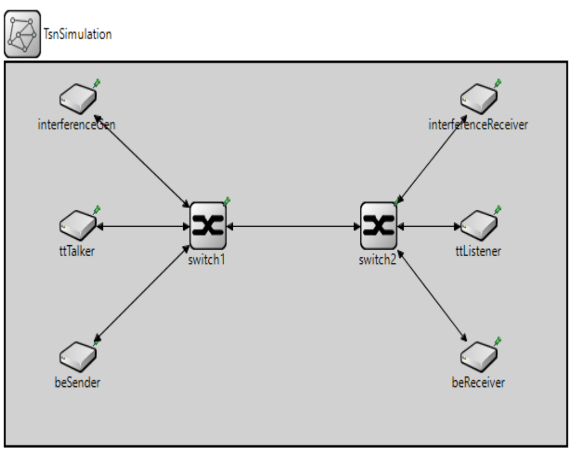

Fig. 3. Simulation Diagram

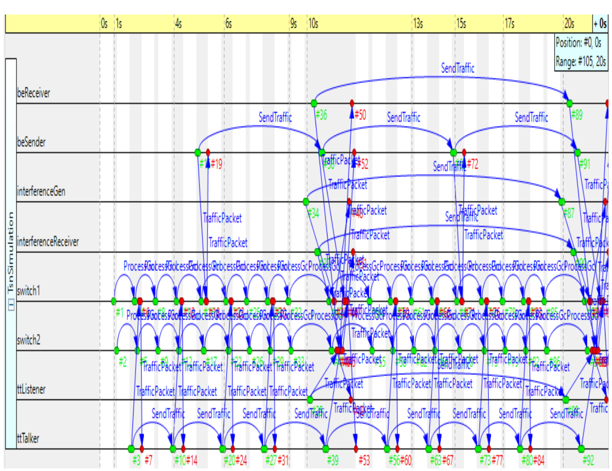

Fig. 4. Simulation Record Diagram

IV. Evaluation Results

The functions were programmed using C++ and the TSN topology was built using OMNET++. The simulation configuration for the TSN talker included parameters such as MAC address, traffic type, size, and send interval. The simulation results were analyzed to calculate the end-to-end latencies for both time-triggered and best-effort traffic. The results indicated that while the end-to-end latency of best-effort traffic was significantly affected by interference traffic, the time-triggered traffic maintained a stable latency, demonstrating the effectiveness of the TSN protocols in guaranteeing deterministic low latency.

V. Conclusion

In this paper, we presented a TSN simulation model developed using OMNeT++. The model features a GCL-driven switch and a representative network topology, along with a method for GCL calculation based on network parameters. The results indicate that the TSN model can guarantee deterministic end-to-end latency, making it a valuable tool for evaluating TSN scheduling algorithms and GCL synthesis methods.

Acknowledgment

A huge thanks to our course teacher Abdullahil Kafi, Assistant Professor, Dept. of CSE, IIUC for supervising and guiding us throughout the project.

References

[1] IEEE, “802.3 Standard for Ethernet,” 2015.
[2] IEEE, “Time-Sensitive Networking http://www.ieee802.org/1/pages/tsn.html, 2016.
[3] IEEE, “802.1ASrev—Timing and Synchronization for Time-Sensitive Applications,” http://www.ieee802.org/1/pages/802.1AS-rev.html, 2017.
[4] IEEE, “802.1Qbv—Enhancements for Scheduled Traffic,” http://www.ieee802.org/1/pages/802.1bv.html, 2015.
[5] IEEE, “P802.1Qcc – Stream Reservation Protocol (SRP) Enhancements and Performance Improvements”, https://1.ieee802.org/tsn/802-1qcc/, 2017
[6] IEEE, “802.1Qci—Per-Stream Filtering and Policing,” http://www.ieee802.org/1/pages/802.1ci.html, 2016.
[7] C. Park, J. Lee, T. Tan, and S. Park, “Simulation of scheduled traffic for the IEEE 802.1 time sensitive networking,” in Information Science and Applications, LNEE Vol. 376. Springer, Singapore, 2016, pp. 75–83.
[8] P. Heise F. Geyer, R. Obermaisser, “TSimNet: An Industrial Time Sensitive Networking Simulation Framework Based on OMNeT++” in IEEE NTMS Larnaca Cyprus: November 2016.
[9] F. Smirnov, M. Glaß, F. Reimann, and J. Teich, “Formal timing analysis of non-scheduled traffic in automotive scheduled TSN networks,” in Design, Automation & Test in Europe Conference & Exhibition, March 2017.
[10] A. M. Kentis, M. S. Berger, J. Soler, “Effects of port congestion in the gate control list scheduling of time sensitive networks,” in the 8th International Conference on Network of the Future, November 2017.
[11] L. X. Zhao, and P. Pop, “Timing Analysis of AVB Traffic in TSN Networks using Network Calculus” in 2018 IEEE Real-Time and Embedded Technology and Applications Symposium, April 2018
[12] S. S. Craciunas, R. S. Oliver, M. Chmelik, and W. Steiner, “Scheduling Real-Time Communication in IEEE 802.1Qbv Time Sensitive Networks,” in Proceedings of the 24th International Conference on Real-Time Networks and Systems, ACM, pp. 183-192, 2016.
[13] “OMNeT++.” [Online]. Available: http://www.omnetpp.org/
[14] “INET Framework.” [Online]. Available: http://inet.omnetpp.org/
[15] “CoRE4INET Framework.” [Online]. Available: https://core4inet.core.org/
[16] “OMNeT++ Simulation Manual.”[Online]. Available: https://www.omnetpp.org/doc/omnetpp/manual/
[17] Junhui Jiang, Yuting Li, Seung Ho Hong, Aidong Xu, Kai Wang ”A Time-sensitive Networking (TSN) Simulation Model Based on OMNET++”