# Time Sensitive Networking for Improvement of Digital Twin Synchronization in Industrial Internet of Things

Elaa Jamazi

BCRG Advisor: Prof. Dr. Berk Canberk

## Introduction

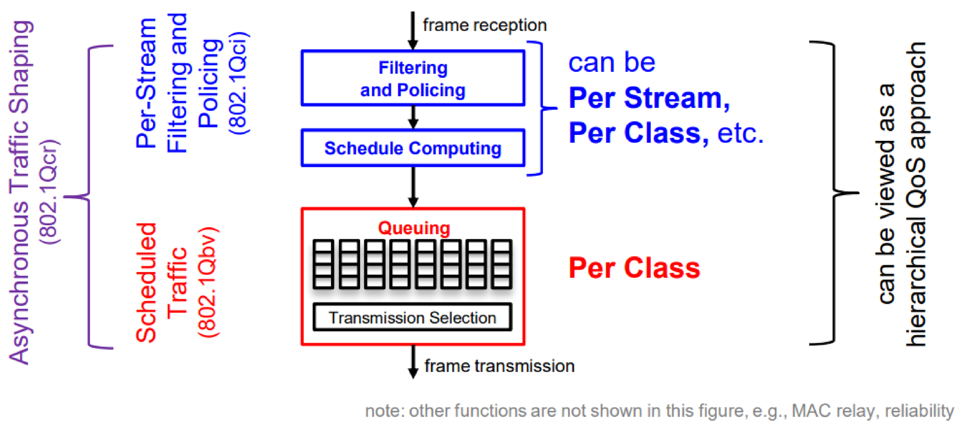
Digital Twin (DT) is a core element of Smart Manufacturing. It allows for real-time monitoring. The data collected by the digital twin can be used for predictive maintainance and optimization, both characteristics of smart manufacturing, a key feature in Industry 4.0. Understandably, this demands a well synchronized system. For this purpose, Time Sensitive Networking can be used.

## Methods

TSN offers multiple protocols and algorithms. They need to be chosen carefully to meet the project needs, without causing overhead. The various options are listed below along with the choices I made for this Project.

### Time Sensitive Networking

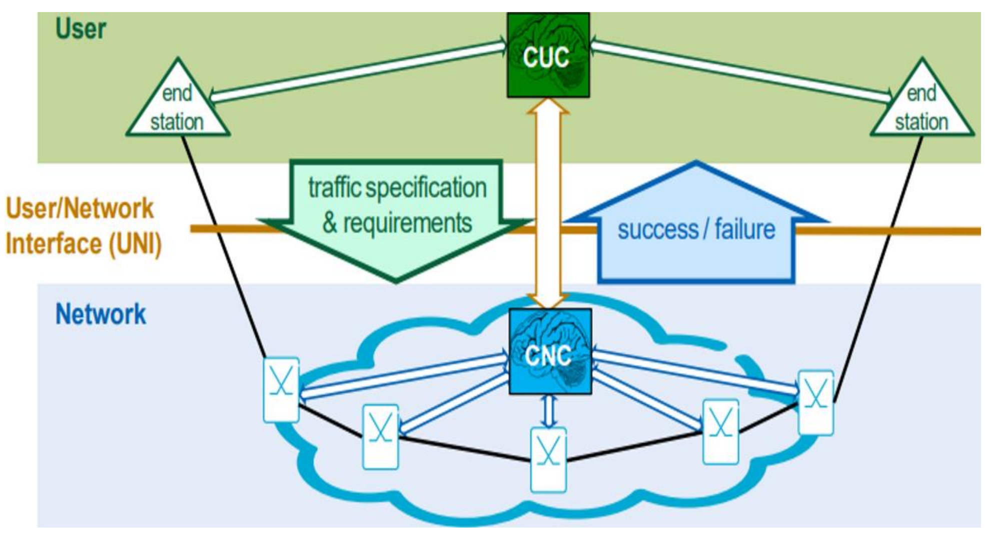
- Time-Sensitive Networking (TSN) standards have been created by the IEEE 802.1 Task Group, which expand Ethernet technology to allow deterministic communication for real-time.
- TSN mixes a variety of traffic kinds with a variety of scheduling [1] algorithms:
  - Time-Aware Shaper 802.1 Qbv
  - Asynchronous Traffic Shaper 802.1 Qcr
  - Credit-Based Shaper 802.1 Qav
- TSN uses Frame Preemption, which establishes a time-critical frame of service that has priority over non-time-critical frames.
- Chosen Traffic Shaper: Asynchronous Traffic Shaper




*Quality of Service Functions for Latency [2]*

### TSN Configuration for resource management

- Centralized Network Configuration (CNC): handles stream reservation centrally, meaning that, , end-stations transmit requests for a certain stream (through edge port) without knowing the network setup
- Centralized User Configurtion (CUC): a logical function that receives requests from TSN endpoints for TSN Flow setup. In turn, the CUC will send TSN Flow configuration data to the CNC



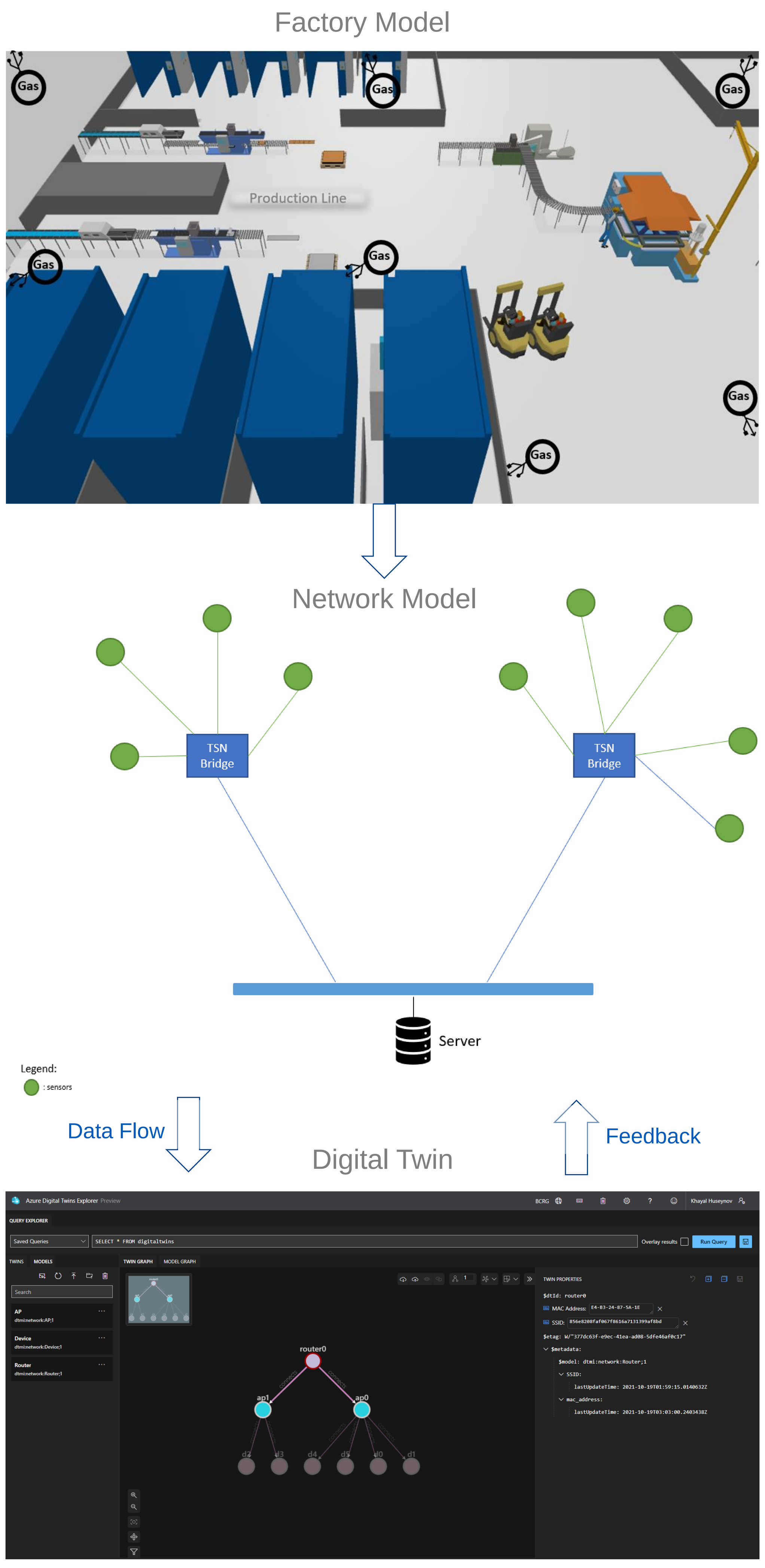

*Fully Centralized Configuration Model [2]*

End stations, their capabilities, and user requirements are discovered by the CUC, and delay-optimized configurations are created. This architecture allows for a simple implementation and thus simple reconfiguration for future update [2]. The end stations can have two roles: ***listener*** and/or ***talker.*** The listener is the station that recieves the TSN frames as for the talker it is the station that sends them. In my scenario, the Digital Twin plays both roles. Listener when recieving frames from the sensors and talker when sending frames after giving decisions regarding the elements of the network. As for the sensors they are talkers since they are only tasked with relaying the current status of the air in the factory. Other listeners in the network maybe ventilators, alarms, etc..

The next section explains the use case with more details.

## Use Case

The scenario chosen to apply this approach is that of a paint factory. There are various air quality sensors placed around the factory. The Digital Twin has a replica of the physical components and constantly recieves data from them. The data represents the concertations of Volatile Organic Compounds such as Ethylene and Methylene, Flue Gas Emissions and some other special emmisions depending on the region in the Factory. These controls are important because they are both govenmently mendated and related to work-place safety. The DT analyzes this data and gives decisions accordingly. If the levels are too high, ventelation may be actived.

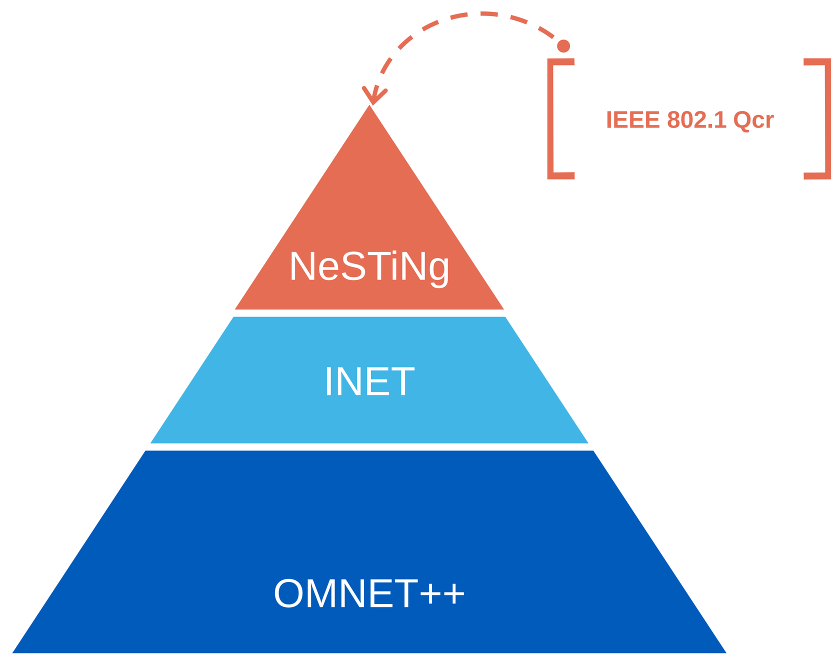



*The proposed scenario*

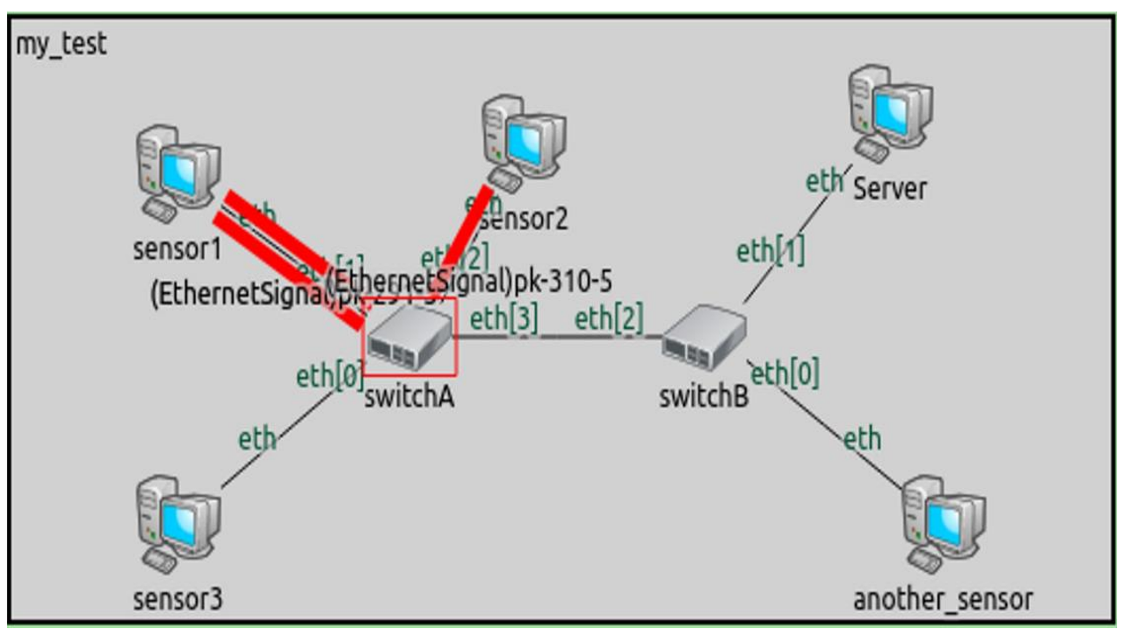
To ensure the previously mentioned scenario, TSN bridges are placed throughout the factory to provide TSN based communication between the DT and the sensors.

## Simulation

The simulation of this network was conducted on OMNET++ Discrete Event Simulator. The open-source model library: INET was used since it provides wireless, wired, and mobile network capabilities and it is suitable for validating and trying new protocols like the TSN specifications and protocols. On top of that, the NeSTiNg [3] framework was utilized.




The next figure shows the structure of the TSN enabled network simulation I conducted. It consists of 4 air quality sensors, a server where our Digital Twin will run, and 2 TSN bridges. During the simulation traffic is generated from the sensors using a data set that contains gas concentrations. This data is forwarded to the server in a deterministic manner thanks to the TSN capabilities of the switches.

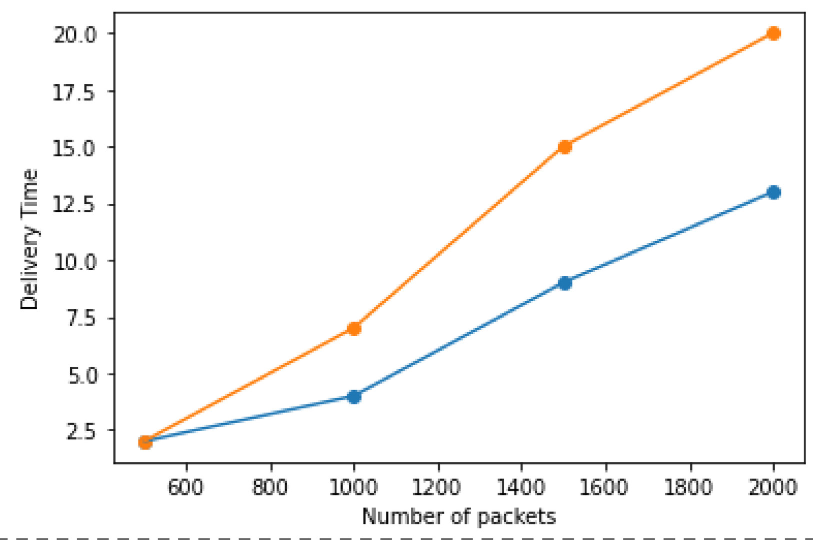



## Results

My evaluation criteria was based on two factors: **Throughput** and **Packet Delivery Rate (PDR)**.

### THROUGHPUT:

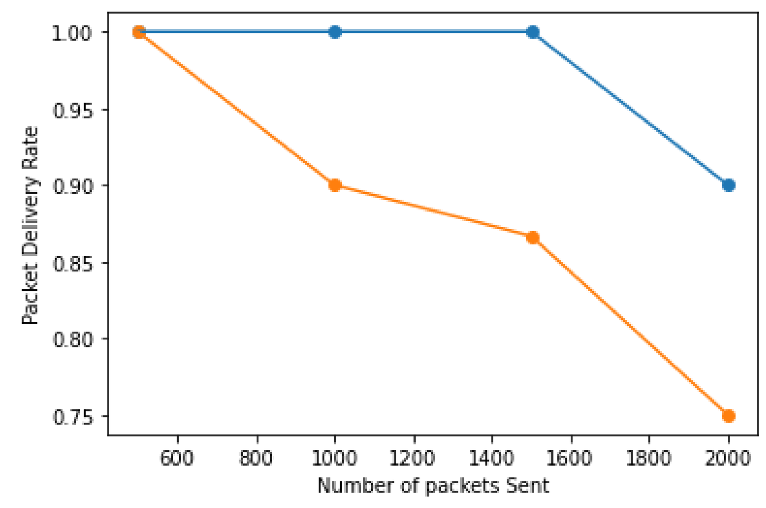
I retrieved data related to the packets delivered to the server from the OMNET++ simulation. From the simulation logs, I could conclude how much time it took the simulator to finish the packet delivery. I delivered packets 4 time to both the TSN and non-TSN implementation. First try I only had 500 packets, second try 1000 packets, third try 1500 packets and last try 2000 packets. The results are shown in the figure below.




### PACKET DELIVERY RATE:

Instead of retrieving the delivery time from the simulation logs, I retrieved the number of packets received at the server. The number of packets to be sent to the server was determined beforehand so I just applied the simple formula:

$$PDR = \frac{Number\ of\ Recieved\ Packets}{Number\ of\ Packets\ Sent}$$




## Conclusion

In conclusion, this project was an effort to improve the synchronization of a Digital Twin used in industry, a paint factory to be specific. This scenario was chosen to focus on the Time Sensitive Networking Approach and the Digital Twin technology. TSN allowed for an improved synchronization through a higher packet delivery rate and throughput. It performed much better compared to networks that did not use TSN. However, this project was conducted on an Ethernet based topology. I tried to venture into wireless TSN but the results where unsatisfactory mostly due to the limitations of the simulator and the used frameworks. In industry, wireless TSN has gone through some advances, but it still falls behind its wired counterpart. This is an important area of work for the future. It may allow to make TSN a widely adopted technology and a key feature of Industry 4.0. Another area where TSN can be improved is its integration with other protocols or technologies. .

## References

1. Y. Yigit, T. Q. Duong and B. Canberk, "Deep Learning-based Traffic Shaper for Time-Sensitive Networking Switches" in IEEE Communications Standards Magazine, December 2022.
2. Time-Sensitive Networking (TSN) Task Group. IEEE802.1. [Online]. Available: https://1.ieee802.org/tsn/ Accessed Apr. 27, 2022
3. A. B. D. Kinabo, J. B. Mwangama and A. A. Lysko, "Towards Wi-Fi-based Time Sensitive Networking Using OMNeT++/NeSTiNg Simulation Models," 2021 International Conference on Electrical, Computer and Energy Technologies (ICECET), 2021, pp. 1-6, doi: 10.1109/ICECET52533.2021.9698580.

İTÜ ISTANBUL TECHNICAL UNIVERSITY
Department of Computer Engineering

Faculty of Computer and Informatics Engineering
İTÜ Ayazaga Campus, 34469, Maslak/Istanbul
bbf@itu.edu.tr
http://bm.itu.edu.tr