Mälardalen University
School of Innovation Design and Engineering
Västerås, Sweden

Thesis for the Degree of Master of Science in Computer Science with Specialisation in Intelligent Embedded Systems, 30 credits

TIME-SENSITIVE NETWORKING IN DIGITAL SUBSTATIONS

Andreas Johansson
ajn19017@student.mdu.se

Samuel Wågbrant
swt19001@student.mdu.se

Examiner: Thomas Nolte
Mälardalen University, Västerås, Sweden

Supervisors: Inés Álvarez, Mohammad Ashjaei
Mälardalen University, Västerås, Sweden

Industry Supervisor: Rahul Gore
Hitachi Energy, Västerås, Sweden

2024-06-03

Abstract

With the advancements in the Industrial Internet of Things (IIoT), new networking requirements and demands are introduced to Substation Automation Systems (SAS) within electrical power grids. The possibility of merging Information Technology (IT) and Operational Technology (OT) traffic on the same network to achieve higher productivity, however, presents new challenges in providing real-time guarantees to OT traffic. Time-Sensitive Networking (TSN) can be a promising solution that allows IT and OT traffic to coexist seamlessly while still providing real-time guarantees for critical applications. Substations act as critical nodes within power grids, and their digitalisation is a crucial element in the energy transition. A digital substation handles International Electrotechnical Commission (IEC) 61850 protocol traffic such as Generic Object Oriented Substation Event (GOOSE), Sampled Values (SV), and Manufacturing Message Specification (MMS), which all have strict timing requirements. The integration of TSN into these substations could improve the handling of this traffic and, consequently, the controllability of power grids.

This thesis investigates the use of TSN in an IEC 61850 process bus, typically implemented in an SAS. A series of simulated test scenarios were developed to evaluate the impact of TSN compared to traditional networking methods. These included configurations using Ethernet, Ethernet with priority queuing, and TSN with Time-Aware Shaper (TAS) and Credit-Based Shaper (CBS). The results indicate that TSN can meet critical timing requirements, reduce jitter, and manage sporadic traffic effectively under high traffic loads. While the TAS scheduler may increase End-to-End delay for periodic traffic, CBS can reduce it for event-based traffic. Furthermore, robust timing guarantees are ensured for the TSN scenarios by providing a feasible schedule for Scheduled Traffic (ST) and a worst-case response time analysis for Audio-Video Bridging (AVB) traffic. This research highlights TSN's potential to improve grid controllability and reliability through enhanced network performance, illustrating its role in the future of resilient grid technologies.

Acknowledgments

First, we would like to thank Michele Luvisotto at Hitachi Energy for creating the opportunity for us to write this thesis and helping out with the fundamental ideas prior to the thesis. Secondly, we want to thank Rahul Gore, who was responsible for supervising this thesis at Hitachi. His guidance as an excellent industry supervisor was instrumental in steering us in the right direction. We are also grateful to our academic supervisors at Mälardalen University: Inés, for her valuable insights into data communication, TSN, and for helping us navigate various tools and schedulers; and Mohammad, for his readiness to assist when needed and for sharing his extensive knowledge. Finally, we want to thank our families, for their endless support through this long journey of studies at Mälardalen. Thank you!

Andreas Johansson & Samuel Wågbrant

Contents

List of Figures

List of Tables

List of Acronyms

AVB Audio-Video Bridging
CAN Controller Area Network
CBS Credit-Based Shaper
CT Current Transformers
EMI Electro Magnetic Interference
FIFO First In, First Out
GCL Gate Control List
GOOSE Generic Object Oriented Substation Event
GPS Global Positioning System
gPTP generalised Precision Time Protocol
HERMES Heuristic Multiqueue Scheduler
HMI Human-Machine Interface
HSR High-availability Seamless Redundancy
ICMP Internet Control Message Protocol
IEC International Electrotechnical Commission
IED Intelligent Electronic Devices
IIoT Industrial Internet of Things
IoT Internet of Things
IS Information Systems
IT Information Technology
kV kilovolt
LAN Local Area Network
MAC Media Access Control
MMS Manufacturing Message Specification
MU Merging Unit
NED Network Description Language
NeSTiNg Network Simulator for Time-Sensitive Networking
NTP Network Time Protocol
OMNeT++ Objective Modular Network Testbed in C++
OPNET Optimized Network Engineering Tools
OSI Open System Interconnection
OT Operational Technology
PCP Priority Code Point
PoC Proof of Concept
PRP Parallel Redundancy Protocol
PTP Precision Time Protocol
PV Process Value
RFI Radio Frequency Interference
RTOS Real-Time Operating System
SAS Substation Automation System
SCADA Supervisory Control and Data Acquisition
SNMP Simple Network Management Protocol
SNTP Simple Network Time Protocol
SoC System-on-Chip
SSH Secure Shell
ST Scheduled Traffic
SV Sampled Value
TAS Time-Aware Shaper
TSN Time-Sensitive Networking
VLAN Virtual Local Area Network
VT Voltage Transformers

1 Introduction

The technical advances in Industrial Internet of Things (IIoT) have given rise to advanced and emerging applications in the industrial domain. This transformation is characterised by the integration of an increasing number of Internet of Things (IoT) devices into industrial networks, leading to expanded network capacities and heightened traffic demands. Time-Sensitive Networking (TSN) is a promising real-time networking solution for industries seeking to meet the aforementioned demand. Digital substations are one of the applications where this demand is crucial. Substations are critical nodes within the power grid, transforming voltage to suitable levels for efficient transmission and distribution of electricity.

In this thesis, we conduct a comprehensive performance evaluation of the message protocols on the International Electrotechnical Commission (IEC) 61850 process bus within a Substation Automation System (SAS) concerning their latency and jitter. The work is conducted by creating models in the Objective Modular Network Testbed in C++ (OMNeT++) framework. The process bus is a key component in delivering data from primary devices in substations to Intelligent Electronic Devices (IED). The IEDs are microcontrollers capable of receiving information from sensors within the electrical substation and performing necessary actions. The messages under investigation are Generic Object Oriented Substation Event (GOOSE), Sampled Value (SV), and Manufacturing Message Specification (MMS) messages. GOOSE messages are event-based messages such as trip commands from IEDs to primary devices, while the SV messages carry periodic, sampled analogue data such as current and voltage measurements. The MMS messages are sporadic and contain real-time process and control data.

Previous works have made efforts to create models for the IEC 61850 standard in OMNeT++ [1]. Similarly, simulation models have also been created for TSN [2], [3]. However, little research has been conducted on the integration of IEC 61850 and TSN through modelling. Docquier *et al.* [4] presents the most relevant works that could be found for the context of this thesis. In their work, the authors have questioned the relevance of TSN in SASs by conducting two case studies utilising OMNeT++ models. In their work, they targeted the worst-case delay for GOOSE, SV, and MMS messages but did not include analysis of jitter. In other research by Docquier *et al.* [5], they specifically investigate the TSN Time-Aware Shaper (TAS) mechanism, suggest traffic mapping of IEC 61850 protocols to TSN traffic classes, and present an analysis model for the worst-case delay. In this thesis, we perform an extensive benchmarking of the IEC 61850 protocols and examine the network performance in an integrated Information Technology (IT) and Operational Technology (OT) network.

1.1 Brief Research Questions and Methods

The research questions we aim to answer with this study are:

> **RQ1:** What are the typical latency levels of IEC 61850 process bus protocol messages allocated in different TSN classes compared to a standard Ethernet network?
>
> **RQ2:** What is the observed jitter of IEC 61850 process bus protocol messages allocated in different TSN classes compared to a standard Ethernet network?
>
> **RQ3:** What is the effect on clock synchronisation by classifying generalised Precision Time Protocol (gPTP) traffic as Scheduled Traffic (ST)?

Approaching this work, the multimethodological research approach for systems development by Nunamaker *et al.* [6] was employed. Initial efforts were made to ensure that relevant knowledge was gathered of the research field via a literature study. With knowledge obtained from the literature study, a conceptual framework was formed to understand the process of system functionality and procedures. To answer the proposed research questions, experiments were performed via simulations which are efficient during initial development and testing because many different parameters can be tested in a fast manner. Finally, the computer simulations were evaluated and analysed to determine if the results were satisfactory or if any earlier parts needed to be redone. The approach was iterative, meaning any of the previous steps could be redone to achieve a satisfactory result.

Several scenarios were created to analyse the effect of integrating TSN to the IEC 61850 process bus. The scenarios established were standard Ethernet, standard Ethernet with priority queueing, and TSN with TAS and Credit-Based Shaper (CBS). All scenarios were tested with low and high traffic loads. The final results showed that all scenarios managed to meet all deadlines with no misses, however with varying jitter levels. The Heuristic Multiqueue Scheduler (HERMES) scheduler used for scheduling ST in the TSN scenarios increased the End-to-End delay compared to non-TSN scenarios. However, the schedule proved that no ST would miss its deadline. The results indicate that TSN can help reduce End-to-End delay of time-critical messages during high traffic loads if traffic is shaped with CBS or help reduce jitter for time-critical messages during high traffic loads by using TAS. The resulting effects of clock synchronisation show that since gPTP messages are classified as ST and therefore scheduled offline it increases the clock synchronisation error compared to non-TSN scenarios. However, the clocks during the TSN scenarios never drift over the 1 µs requirement defined in the 802.1AS standard.

1.2 Thesis Outline

The structure of the report is as follows: Section 2 provides an overview of substations from an evolutionary perspective and introduces technologies relevant to the modern-day digital substation. Section 3 offers a review of the state of the art and contextualises the work in this thesis. Section 4 presents the motivations and objectives of this research. Section 5 details the methodological approach adopted for the study. Section 7 describes the design and implementation of the work. Section 8 presents the obtained results. Section 9 discusses the results and their implications. Finally, Section 10 provides the final conclusions of the thesis.

2 Background

This section provides an insight into the background of the thesis. Section 2.1 presents a high-level overview of what electrical substations are, their role in the power grid, and the evolution of SASs. Section 2.2 presents an overview of data communication in substations. Section 2.3 details the relevant parts from the IEC 61850 standard. Section 2.4 presents how time synchronisation is handled in substations. Finally TSN and its relevant features are covered in Section 2.5.

2.1 Electrical Substations

An electrical substation is a key part of the power grid system that consists of power generation, transmission and distribution. In power generation, electricity can be produced either using fossil fuels, nuclear power, or renewable sources such as hydro, solar or wind. The transmission system can then carry high-voltage electricity over great distances where the high voltage is used to reduce energy loss. Finally, the distribution system branches electricity to the end users. The electrical substation's role is to transform voltage from high to low and vice versa using transformers [7], [8]. The key components in a substation besides transformers are:

- **Busbars:** Conductors that carry current within the substation. Connects incoming and outgoing circuits as well as transformers and other equipment.
- **Isolators:** Switches that can isolate equipment from the power supply. Ensures that equipment can be maintained and repaired.
- **Circuit breakers:** Switches that protect equipment from damage from overloads or short circuits.

There are different types of substations such as transmission substations and distribution substations [7], [8]. When power is generated, transmission substations are employed to convert voltage to a level that is efficient for transmission over long distances. The voltage usually ranges between 100 and 600 kilovolt (kV) depending on geographic location. There are several benefits to stepping up the voltage during transmission. Since power loss and voltage drops are proportional to the current flowing through the transmission lines the current can be reduced if the voltage is increased. Hence power loss and voltage drops are also reduced. Furthermore, increasing the voltage can also reduce costs since transmission lines can be reduced in size since their size are proportional to the maximum current they can carry. The electricity is typically delivered to a distribution substation where the voltage is stepped down and distributed into a local area. A distribution substation typically steps down the voltage to 11 kV, but can be either lower or higher depending on the size of the area and power needed. For the power to be utilised in homes, the voltage is further lowered to around 120-400 volts depending on location and standards. Figure 1 shows a general model of how substations are used in the power grid. In the figure, voltage is switched up in (2), close to the point of generation, and then switched down in (4), close to the point of consumption.

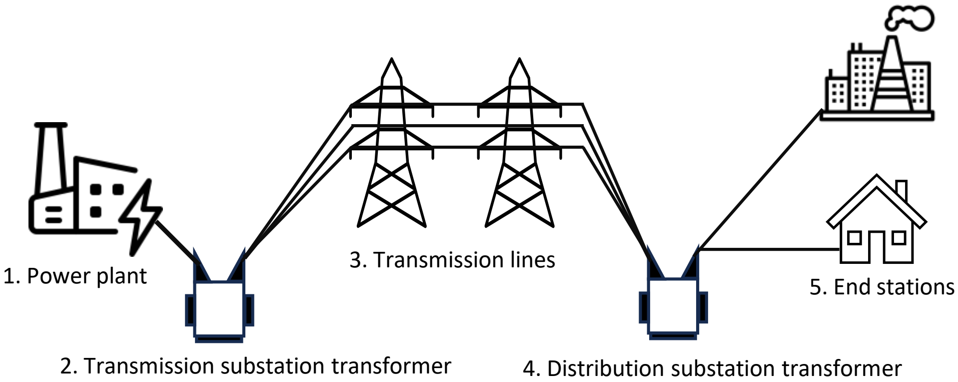

Figure 1: Substations in the power grid

The evolution and technological advancements in substations have been significantly influenced by several factors, such as increasing demand for uninterrupted, reliable power supply as well as environmentally friendly technologies [9], [10]. Automation has emerged as a key factor in ensuring an efficient and dependable power supply, leading to the development and necessity of SASs. The earliest SASs were hardwired and data communication was established through the use of copper wires. Each device was tasked to perform a single monitoring or control function. In around 1995, the digital SAS emerged. Digitalisation in SASs was accomplished by substituting traditional hardwired communication with digital communication links and buses, greatly reducing the amount of cabling needed. This shift facilitated the integration of devices and enhanced interoperability among different systems. In 2005, the IEC 61850 standard further advanced the functionality of SASs by enabling more sophisticated data management and processing capabilities. Furthermore, the use of standardised communication protocols allowed for better interoperability of systems between different manufacturers. 2015 saw the implementation of both a station bus and process bus in SASs, according to the IEC 61850 standard. This adaptation led to a surge in digitalisation and the amount of devices that could be connected to the network.

Figure 2 shows a general depiction of the substation architecture which will be covered in more detail in Section 2.2. The station level of an SAS is utilised to monitor, control, and protect via Human-Machine Interface (HMI) and Supervisory Control and Data Acquisition (SCADA) systems [11]. This allows for human interaction and supervision of the substation and data is typically gathered from the bay level via the station bus. On the other hand, the process level collects and processes data from primary devices such as Voltage Transformers (VT) and Current Transformers (CT) via the process bus.

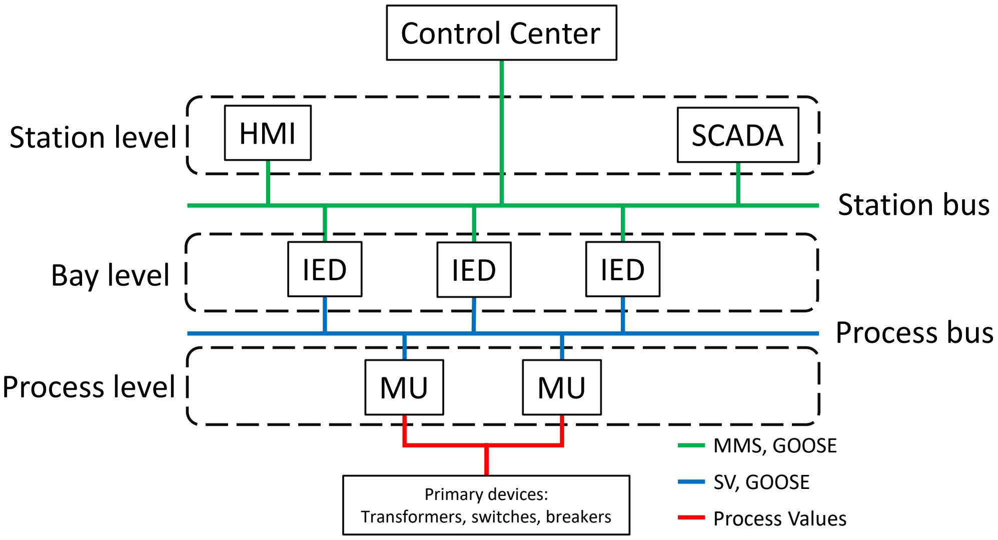

Figure 2: General architecture and communication scheme of a SAS

2.2 Data Communication in Substations

The purpose of a SAS is to monitor, control, and protect systems within an electric substation [9]. To perform these tasks, SASs are typically formed by several IEDs. IEDs are microcontrollers capable of receiving information from sensors within the electrical substation and performing actions if deemed necessary [12]. An IED can be found in various positions throughout the electrical substation, depending on the role and functionality it serves [9]. To accommodate the increased demand for electrical substations which can sustain higher operating voltages, additional IEDs are critical to sustain the need for monitoring, protection, and control. To facilitate these needs, data communication is required within the electrical substations to enable communication between the devices found in the SAS.

To support the communication needs of all devices found in the substation, the substation is equipped with two primary communication buses [13]. Specifically, data in the substation will travel on either the station bus or the process bus depending on its origin. An SAS is often depicted in a hierarchical structure with three different levels, the station, bay, and process levels. These three levels are connected using the station bus and process bus, see Figure 2. An IED can be found in all three levels with varying functionalities depending on its placement within the substation [9]. The Merging Unit (MU) is typically installed close to the primary devices in the switchyard and is responsible for aggregating Process Value (PV) data and converting analogue values, e.g. current and voltage levels, to digital values. IEDs found in the bay level utilise the digital values converted by the MU to monitor system values. The process bus enables the communication between the MU in the process level and the IEDs found in the bay level [14]. It is responsible for carrying the time-critical event-triggered GOOSE messages and the time-triggered SV. To ensure the integrity of the data, the process bus is most often an optical link [15]. An optical link is often preferred for its high throughput and innate characteristic of being resistant against Electro Magnetic Interference (EMI) / Radio Frequency Interference (RFI). Even though optical links require high integration costs, the benefits are deemed to outweigh the drawbacks.

2.3 IEC 61850 Standard

The IEC 61850 standard is a framework that focuses on the design and operation of data communication within electrical SASs [16]. This includes the use of standardised communication protocols to exchange data between IEDs, including messages such as MMS, GOOSE, and SV. Furthermore, standardised network architecture, data modelling, naming conventions, and services facilitate interoperability between devices from different manufacturers, reducing the cost and complexity of integrating systems. Specifically, the data modelling is object-oriented and hierarchical which makes it easy to manage and share data between IEDs. Additionally, the IEC 61850 standard is designed to take advantage of modern networking technologies such as Ethernet in order to provide high-speed interfaces for connecting devices to the network. Several types of traffic messages are defined in IEC 61850 with the most notable being:

- **SV messages -** Used to transfer aggregated current and voltage measurements from MUs. The traffic is periodic with a fixed sampling frequency.
- **GOOSE messages -** Sporadic event-based messages used for protection and control applications such as status changes or alarm notifications between IEDs.
- **MMS messages -** Used for client-server communication between IEDs. Messages are sporadic and can be used for sending commands, status updates, and transferring files.

Refer to Table 1 and Table 2 for a general overview of traffic types that reside in SASs [17]. Table 1 presents a general overview of traffic types that are defined by IEC 61850, while Table 2 shows other traffic types that are commonly found in substation networks. These traffic types can correspond to the types supported by other protocols, such as High-availability Seamless Redundancy (HSR), Parallel Redundancy Protocol (PRP), Internet Control Message Protocol (ICMP), Secure Shell (SSH), Simple Network Management Protocol (SNMP) and others.

IEC 61850 Traffic	Ethertype or protocol	Periodicity	Has deadline?	Criticality
GOOSE 1A (Multicast)	88-B8 / UDP/IP	Sporadic	Yes	High
GOOSE 1B (Multicast)	88-B9 / UDP/IP	Sporadic	Yes	Medium
SV (Multicast)	88-BA / UDP/IP	Cyclic	Yes	High
MMS medium speed	TCP/IP	Sporadic	No	Low
MMS Low speed	TCP/IP	Sporadic	No	Low
MMS file transfer	TCP/IP, FTP	Sporadic	No	Low
Time synchronisation	SNTP, PTP	Cyclic	Yes	High
Command (MMS)	IP	Sporadic	Yes	Low

Table 1: IEC 61850 traffic types

Other Traffic	Ethertype or protocol	Periodicity	Has deadline?	Criticality
IEC 60870-5- 104	TCP/IP	Sporadic	No	Low
DNP3	UDP/IP or TCP/IP	Sporadic	No	Low
Mgmt. Monitoring	TCP/IP	Sporadic	No	Low
Audio/Video	UDP/IP	Cyclic or sporadic	No	Low
Other	TCP/IP, FTP	Sporadic	No	Low

Table 2: Other traffic types commonly found in substation networks

2.4 Time Synchronisation in Substations

A SAS requires precise synchronisation between devices to ensure the reliability, safety, and efficiency of the system, and in extension, of the electrical power grid as well [13], [18]. Within IEC 61850 and the domain of substation automation, some commonly used time synchronisation protocols today are Precision Time Protocol (PTP), Network Time Protocol (NTP), Simple Network Time Protocol (SNTP), and Global Positioning System (GPS).

- **PTP**, which is also known as IEEE 1588, is a protocol used for high-precision synchronisation applications. It is accurate down to the microsecond range and is particularly useful in mission-critical applications. For example, protection devices such as relays and breakers need precise timing to detect and respond to faults. Synchronisation ensures that the system can operate in precise coordination, ensuring safety. This type of accurate synchronisation is useful and commonly found in the process and bay level of a SAS.
- **NTP** and its light version **SNTP** are used for general application purposes within substations where requirements of synchronisation are more relaxed. The synchronisation accuracy of NTP and SNTP is lower compared to that of PTP.
- **GPS** used to be a widely adopted synchronisation technique for SASs. However, due to its installation costs, maintenance needs, and failure modes, PTP is often the preferred method of synchronisation today.

2.4.1 PTP Synchronisation

The IEEE 1588 standard (PTP) [19] defines a method to synchronise clocks throughout a computer network that requires precise timing. It achieves synchronisation by exchanging timing messages between a master clock and slave clocks over the network [20]. Figure 3 shows a representation of the synchronisation message exchange. The master begins by sending a *Sync* message to a slave and immediately following up with a *Follow_Up* message that contains the exact timestamp of the sync message (T1). The slave performs a timestamp on the time of arrival of the sync message (T2). The *Delay_Req* and *Delay_Resp* messages are used to calculate the round-trip time to the master clock. After the exchange is done, the slave clock knows all the T-values and performs the following calculation to learn its synchronisation offset:

$$Offset = \frac{(T_2 - T_1) - (T_4 - T_3)}{2}$$

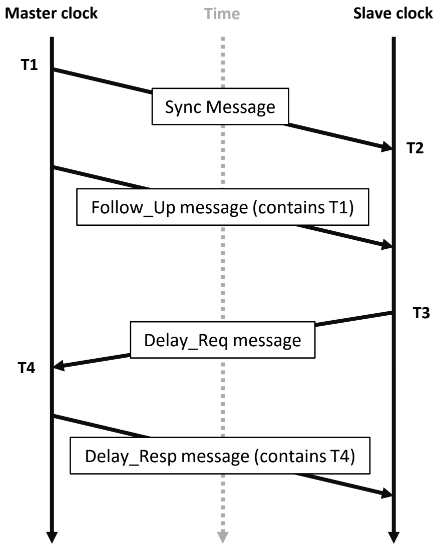

Figure 3: PTP synchronisation mechanism

2.5 Time-Sensitive Networking

TSN is a set of standards developed by the IEEE 802.1 TSN task group[1] which aims to provide real-time guarantees for time-critical data to travel over standard Ethernet in a deterministic manner [21], [22]. Based on the widely adopted Ethernet standard, TSN offers ease-of-integration and a cost-effective solution for systems already using Ethernet communications, contributing to its growing popularity in recent years. TSN manages to provide real-time guarantees by expanding and providing new standards upon the existing Ethernet standards [23]. TSN was originally developed and maintained by the Audio-Video Bridging (AVB) task group which later led to the foundation of the TSN task group in 2012. Therefore, there are several standards developed and defined by the AVB task group which are still in use and considered a part of the TSN standard. The following sections will provide an initial overview of the Ethernet standards which TSN is based upon and thereafter detail the TSN standards used within this thesis.

2.5.1 Ethernet Standards Used Within Time-Sensitive Networking

The IEEE 802.1 TSN task group defines six different Ethernet standards which are the base standards for TSN. The following standards have later been expanded and modified to create the standards which define TSN.

- **IEEE Std 802.1Q-2022 Bridges and Bridged Networks [24]:** The standard which defines how Bridged Networks shall support Media Access Control (MAC) services, and the operation of MAC bridges and Virtual Local Area Network (VLAN) bridges.
- **IEEE Std 802.1AB-2016 Link Layer Discovery Protocol (LLDP) [25]:** The standard which defines how managed objects in a network can discover the topology and neighbouring objects within a Local Area Network (LAN).
- **IEEE Std 802.1AS-2020 Timing and Synchronisation for Time-Sensitive Applications [26]:** The standard which defines how to achieve time synchronisation within a LAN using PTP. This includes the transit of the synchronised time, how to detect unsynchronised parts of the network, and the selection of the timing source.
- **IEEE Std 802.1AX-2020 Link Aggregation [27]:** The standard defines how several physical links can be aggregated into one logical link. This helps increase bandwidth, load-sharing, and redundancy across the network.

---

[1] https://1.ieee802.org/tsn/

- **IEEE Std 802.1CB-2017 Frame Replication and Elimination for Reliability [28]:** The standard which defines how packets travelling over bridges should supply identification for redundant transmissions to avoid duplicate packets and how redundant packets can be disregarded safely.
- **IEEE Std 802.1CS-2020 Link-local Registration Protocol [29]:** The standard which defines how entries should be registered and deleted in a database from one end of a point-to-point link to the other.

2.5.2 Time Synchronisation

For TSN to correctly function and be able to sustain its deterministic behaviour, synchronisation of clocks is required [21]. The synchronisation of clocks enables packets to be distributed in a way that reduces latency and jitter, thereby ensuring that deadlines are met for time-critical data. To achieve synchronisation, TSN adopts the IEEE Std 802.1AS-2020 [26] by utilising gPTP which is a scaled-down profile of PTP which is described in Section 2.4.1. Both protocols function similarly, however, there are some minor differences to ensure gPTP's adaptability for time-critical applications [26, pp. 46-47]. One of the main differences is how the measurements are gathered. In PTP, a measurement is done by calculating the round trip from the master clock to a slave clock. In gPTP on the other hand, these calculations are done in a hierarchical fashion between neighbor systems. See Figure 3 and 4 for a comparison of the two standards. With four known separate time stamps, the delay and offset can then be calculated with the following formulas [26, p. 43].

$$Delay = \frac{(T_2 - T_1) + (T_4 - T_3)}{2}$$

$$Offset = \frac{(T_2 - T_1) - (T_4 - T_3)}{2}$$

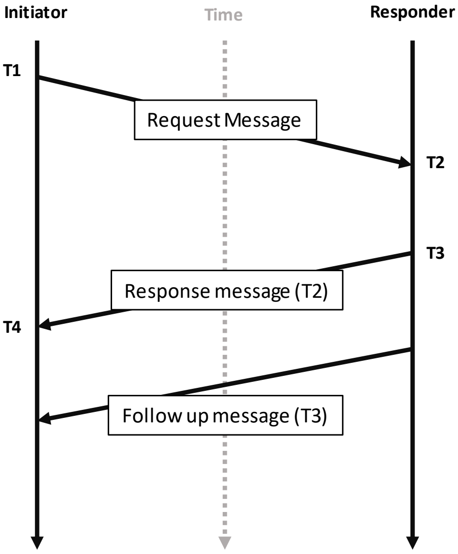

Figure 4: gPTP synchronisation mechanism

2.5.3 Time-Aware Shaper

TAS provides deterministic scheduling for time-triggered traffic, ensuring latency and jitter requirements are met [30]. TAS is defined within IEEE Std 802.1Qbv and will schedule traffic based on pre-defined classes. Depending on the class to which the traffic belongs, it will be placed in one of eight priority queues. The definition of the eight priorities is done with the VLAN ID tag of the 802.1Q frames, ranging from 0-7. To ensure that time-critical traffic is transmitted within the deadline, a Gate Control List (GCL) manages which queues are permitted to transmit at a given time, see Figure 5. With the GCL operation, time-critical traffic will have reserved time slots to ensure that it is received within the deadline. To protect time-critical traffic from being blocked by low-priority traffic, e.g. low-priority traffic starts sending before the time slot of the time-critical traffic and has not completed its transmission before the start of the time-critical traffic slot, guard bands are implemented. Guard bands will ensure that no traffic starts sending on the transmission line before the time slot of time-critical data. The length of the guard band is the time it takes to transmit a maximum-size frame over the transmission link. This ensures that any low-priority packet sending will be able to finish its transmission during the guard band period. For these features to operate as intended, synchronisation of clocks is critical for TAS. If devices on the network are not synchronised with each other, the gates of devices in the network might open at different points in time. Traffic can then be stuck at parts of the network due to gates being closed, which in turn can lead to missed deadlines.

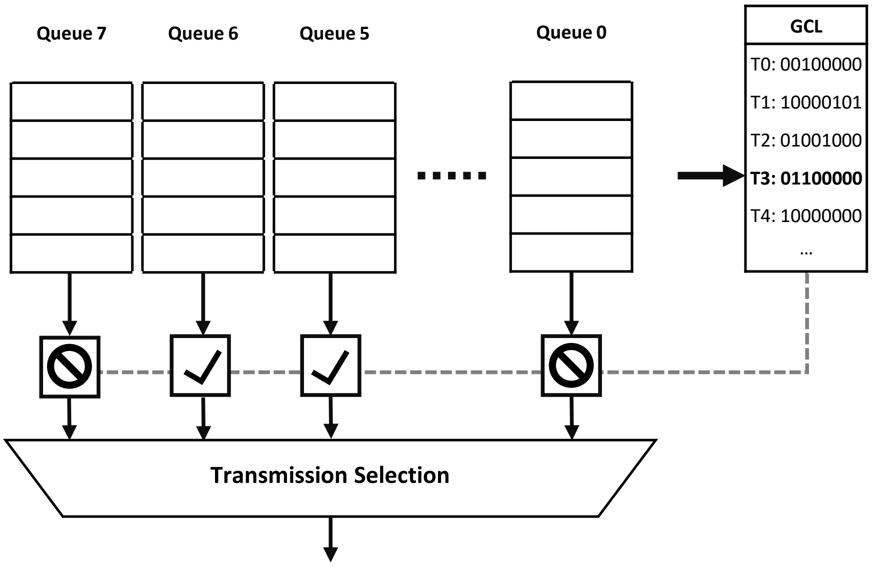

Figure 5: TAS algorithm operation at T3

2.5.4 Credit-Based Shaper

CBS is a shaper within TSN designed to limit bursts and reduce buffering of traffic [31], [32]. CBS is defined within IEEE Std 802.1Qav and was originally developed for AVB systems to manage the strict timing requirements for audio and video streaming. The standard describes how it can be applied to two of the eight traffic classes, *Class A* and *Class B*. Each of the two AVB classes will have a dedicated credit which can take the state *idle*, *sending*, *stopped*, or *replenishing*. Traffic can be sent when either of the two classes has a credit of zero or higher, if they have the same credit value *Class A* will have priority. When a class is sending, the correlated credit will be in the *sending* state and reduced at a rate defined as the *sendingSlope*. When a class is not allowed to send, the credit will be placed in the *idle* state and gain credits at a rate defined as the *idleSlope*. If traffic is blocked by higher priority traffic, it will be placed in the *stopped* state and the credit rate will be frozen. If a traffic class has no traffic to send it will reset the credit value to zero if it previously had a positive credit or it will be placed in the *replenish* state and gain credits until it reaches zero.

2.5.5 TAS and CBS cooperation

When implementing TSN, a common practice is combining several shapers on the same egress interface to accommodate different traffic types [33]. TAS allows for the scheduling of high-priority traffic with guarantees. However, scheduling all high-priority traffic using TAS can increase complexity of the system. It is therefore recommended to use additional shapers for scheduling of high-priority non-ST traffic. If TAS is used with CBS, TAS acts as the primary controller for all gates. During the transmission of ST, all queues except the one for ST will be closed and if no ST is scheduled all other gates will open, allowing CBS to operate and manage the non-ST queues. Figure 6 illustrates the combined use of TAS and CBS. Here, CBS manages queue 5-6 and operates only when TAS is not scheduled to transmit ST.

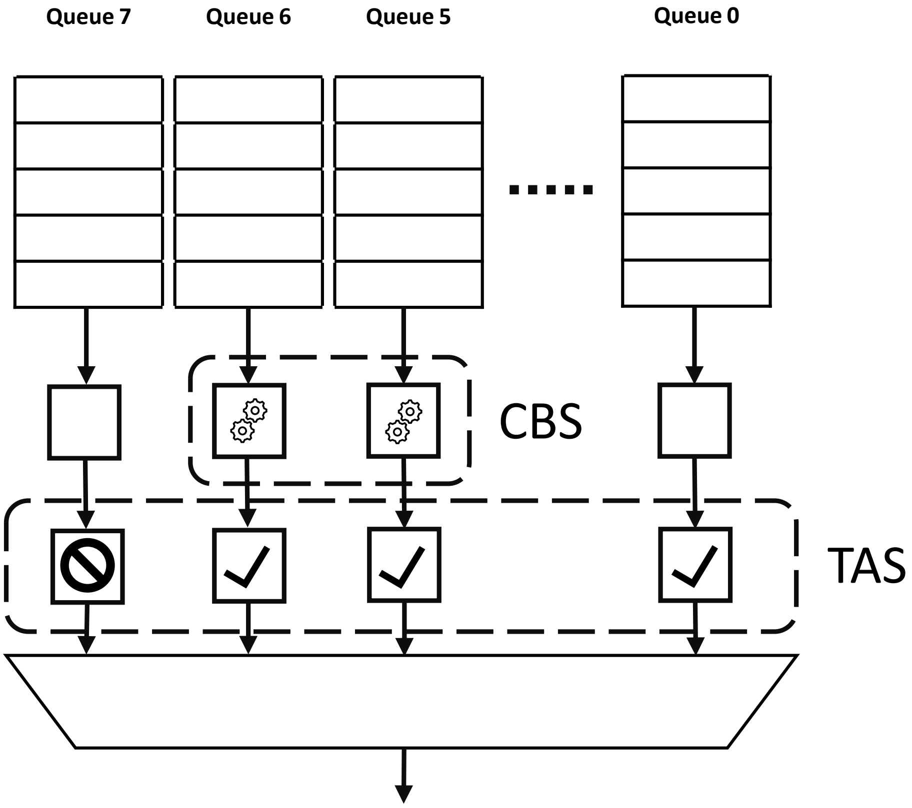

Figure 6: TAS and CBS used together

3 Related Work

This section explores the relevance of TSN in a SAS setting with a comprehensive analysis of works related to real-time communication in digital substations. Our focus is mainly on simulation modelling for performance evaluation, parallelling the approach of this thesis, but we also consider studies involving hardware experiments to provide a broader perspective. The selection of these works was done by evaluating their contributions to the field and understanding the complexities and challenges associated with implementing TSN in digital substations.

3.1 Modelling of TSN and IEC 61850

To evaluate a digital substation's traffic performance when implementing TSN, it is crucial to identify and consider suitable modelling tools. Significant efforts have been made in creating functional models for the simulation of both TSN and IEC 61850. However, research focusing on the integration of IEC 61850 with TSN is still limited. The primary tool used in performance evaluation research for TSN simulation is the OMNeT++ simulator. Researchers have implemented modules within this framework to simulate TSN, such as Network Simulator for Time-Sensitive Networking (NeSTiNg) [2]. Additionally, there have been ongoing efforts since 2022 to incorporate TSN into the OMNeT++ INET framework [3]. This work is a collaboration between OMNeT++ developers and the automotive industry.

Efforts have also been made in creating IEC 61850 models that extend OMNeT++, such as the work by Leon *et al.* [1]. The authors addressed the lack of adequate simulation models for assessing the End-to-End delay of GOOSE and SV messages, which are critical for the operation of electrical power systems. Leon *et al.* proposed a range of simulation models, including those for IEDs and MUs. Additionally, they developed a switch model that complies with Ethernet and VLAN standards, incorporated VLAN tagging, and, importantly, included models for GOOSE and SV messaging. The accuracy of these models was validated using real IEC 61850 equipment. The difference in simulated and experimental maximum End-to-End delay varied between 0.6 - 5.2 µs and 1.2 - 13.7%, depending on the message source. The experimental results consistently showed a smaller End-to-End delay than the simulated ones, suggesting that successful simulations are likely to succeed in real hardware experiments. The authors concluded that the developed simulation models effectively represent communication processes in electrical substations. Furthermore, their work, which is published and free to use, has spurred further research, such as the work by Docquier *et al.* [34]. Here, the authors identified and bridged a gap in substation configuration and Ethernet performance models. Their main contribution was the development of a tool that automatically generates simulation models from description files, which can then be used for performance evaluation.

Out of the reviewed papers, another popular tool for simulating IEC 61850 is Optimized Network Engineering Tools (OPNET), as demonstrated in [35], [36], and [37]. However, the trend for combining both IEC 61850 and TSN simulations seems to be leaning towards the use of OMNeT++.

3.2 The relevance of TSN for Substation Communication

Docquier *et al.* [4] focuses on the evaluation of TSN and its relevancy in the context of substation communication. The authors mention that the IEC 61850 standard specifies several protocols based on Ethernet but has recently started moving towards TSN which prompted them to raise a critical question in this shift: "*Is TSN always relevant for substation communication networks?*". The study delves into this question by performing two case studies in which they highlight the benefits and drawbacks of switched Ethernet and TSN. The two case studies were carried out by doing performance evaluations utilising OMNeT++ together with NeSTiNg and IEC 61850 modules from [1] and [34]. Two distinct case studies were presented:

(1) A star topology with one central switch and five adjacent switches, each switch then connecting three to ten end devices.

(2) A line topology comprising five switches connected in series, each with one to three end devices.

The key focus in the study [4] was to measure the worst-case delay metric of GOOSE, SV, and MMS messages. Simulations ran for 1 second and were repeated 100 times with varied seeds. In the first case study, which involves a larger number of end devices and a heavy traffic load for the central switch, the findings reveal that while TAS underperforms in managing real-time traffic requirements for GOOSE and MMS, it shows an improvement in SV performance. Contrarily, in the second case study, which features a different network topology, all traffic types, including GOOSE, MMS, and SV, successfully meet their real-time requirements, with SV showing a marked improvement in traffic performance using the TAS. The paper suggests that the decision to use TAS depends heavily on the number of egress ports involved in high-priority traffic. The authors highlight the complexity of the problem and recommend thorough investigation before implementing TSN with TAS using either of the described approaches.

3.3 Simulating and Analysing an IEC 61580 Environment

Kanabar *et al.* [35] conducted performance evaluations of the IEC 61850 Process Bus in 2011 using OPNET. In their work, they modelled a ring topology, commonly found in SAS architecture. The topology was based on a typical 345/230 kV transmission substation containing a total of 20 CTs and 8 VTs. In their study, they deliberately introduced faults in order to explore methods for implementing corrective measures, specifically targeting worst-case scenarios. By considering parameters such as sampling frequency, background traffic, and buffer size in switches the authors showed that the parameters influenced SV delays and packet loss. Furthermore, Kanabar *et al.* proposed an SV estimation algorithm to alleviate the experienced problems. They also discussed the possible use of adaptive filtering based on phasor estimation using a least-squares approach, however, they concluded it was not feasible due to its applicability only for scenarios involving the loss of one sampled value.

In [38], Leon *et al.* build upon their previous work in [1] where they created IEC 61850 models for OMNeT++. The focus of their new paper was to perform real-time analysis of time-critical messages within a digital substation. They developed an analytical model by leveraging classical real-time scheduling theory for evaluating the timing characteristics of SV and GOOSE messages. The evaluation was done by performing simulations, applying the analytical model and validating the simulations on IEC 61850 hardware. The main conclusions of the study were that their proposed model was effective for timing assessment and enabled a systematic approach to the schedulability analysis of a process bus, even under high-load scenarios.

Golshani *et al.* [39] modelled and evaluated a SAS environment using OMNeT++ utilising the INET framework. In their work, they created their own IEC 61850 models, and framework, to simulate a whole substation communication network consisting of all three levels of a SAS, as described in Section 2.2 and Figure 2. Their model consisted of one transformer bay, two feeder bays, a server, and a station PC, all connected in a star topology. The feeder bays consisted each of two protection and control IEDs, one circuit breaker, and one MU, while the transformer bay consisted of two protection and control IEDs, two circuit breakers, and one MU. In their analysis, the End-to-End analysis of SV and GOOSE messages was measured between the various units and compared over a 10 Mbps and a 100 Mbps LAN, respectively. In their results, no messages exceeded their specified timing requirements and the authors concluded that the IEC 61850 standard is effective for substation communications, but further refinements and explorations into alternative architectures are needed for enhanced performance and reliability.

3.4 Evaluating TSN in IEC 61580 Environments

Other than the research conducted by Docquier *et al.* [4] which is detailed in Section 3.2, there has been relatively limited research on integrating TSN into digital substations. This topic, however, has been discussed in the IEC Technical Report 61850-90-13 [17], which specifically focuses on possible deterministic networking technologies for IEC 61850.

In this section, we will present two works that are closely related to this thesis in which we aim to simulate and conduct performance evaluations of the IEC 61850 process bus by applying TSN protocols. The first study [5], is a precursor of [4] by Docquier *et al.* in which the authors perform traffic mapping and delay analysis of GOOSE traffic by evaluating the TSN TAS via simulation. The other highly relevant work for this thesis is made by Sanchez *et al.* [40], in which the authors present a TSN Proof of Concept (PoC) integration on hardware in a real-life digital substation.

In [5], the authors contribute to the field by suggesting a mapping between IEC 61850 protocols and TSN traffic classes by considering the characteristics of both standards. The proposed mapping can be observed in Table 3. In their second step, they developed a new worst-case delay due to specific challenges with the combination of IEC 61850 and TSN citing the complex traffic patterns of IEC 61850 substations with its specific requirements on each message type. Furthermore, this real-time analysis method was then used to analyse the worst-case delay of GOOSE messages and compare it to simulated results. The simulation environment included constant sizes for GOOSE 1A/1B and MMS messages, and messages were synchronised to create the worst-case workload. Five scenarios were defined to assess the analytic results, with GOOSE 1A message sizes set to 273B and 325B to test different configurations.

The results in [5] showed that SV traffic was not disturbed by other traffic, giving it zero-jitter and perfectly bounded latency. Furthermore, GOOSE 1A traffic did not experience a delay peak such as GOOSE 1B and MMS, due to its higher priority and lighter bandwidth requirement. The authors noted that the gap between the analytic worst-case delay and the simulated one may be attributed to the low number of simulations and the rarity of certain events occurring. The difference in computed and simulated results can be observed in Table 4.

The main difference between [5] and this thesis is that we will utilise CBS, offer a more detailed analysis of jitter, apply a different scheduling approach, and worst-case delay analysis. Additionally, we will introduce new INET models for IEC 61850 protocols in OMNeT++.

IEC 61850 protocol	TSN class	Ethernet Priority
SV	protected	7
GOOSE 1A	non-protected	6
GOOSE 1B	non-protected	5
MMS	non-protected	4
Best-Effort	non-protected	0-3

Table 3: Proposed mapping of IEC 61850 - TSN by Docquier *et al.*

	Computed WCD (ms)	Simulated WCD (ms)	Gap (%)
Scenario 1	0.840	0.810	3.57
Scenario 2	0.953	0.869	8.81
Scenario 3	0.512	0.500	2.34
Scenario 4	0.529	0.463	9.57

Table 4: Difference between computed and simulated worst-case delay of IEC 61850 traffic over TSN by Docquier *et al.*

Unlike most studies covered so far, Sanchez *et al.* [40] integrated TSN in a real-life hardware setting. The study aimed to present a use case by pairing TSN with substation communication over an Ethernet-based infrastructure, demonstrating the integration of data flows and message types in a digital substation, particularly focusing on critical flows. In their experiments, they utilised two TSN nodes: one programmable System-on-Chip (SoC) that could serve as a TSN Listener/Talker, and one TSN switch. The tests were conducted at the bay level of an SAS to investigate the aforementioned traffic flows, as illustrated in Figure 2. They connected their experimental equipment to substation equipment and subsequently measured the performance of the TSN network by transmitting critical signals. The authors noted some limitations in their work, such as a limited number of activations in the substation environment, with their equipment being safely activated only a small number of times before causing excessive wear. Additionally, the queue buffers used were only 4 kB, which, as the authors highlighted, significantly impacted network traffic. The main results and conclusions of this study were the positive feasibility of using TSN in a real-world substation setting. Critical events originating from the line protection unit could be delivered within 30 microseconds using the TSN system, approximately 170 microseconds faster than the legacy system. Finally, the authors concluded that TSN performance is largely determined by configuration parameters set by the user, which can significantly impact message packet loss and/or latency.

4 Problem Formulation

The technical advances in IIoT have given rise to advanced and emerging applications in the industrial domain [4]. The emerging applications add a large number of IoT devices to industrial networks. As industrial networks expand and network traffic increases, the demand on network capabilities also grows. TSN is a promising real-time networking solution for industries that are seeking to meet the aforementioned demand. One of the main advantages of TSN lies in its foundation on the widely adopted Ethernet communication protocol. Ethernet is known for its high throughput capabilities, enabling TSN to outperform other real-time protocols such as Controller Area Network (CAN) and FlexRay. Furthermore, another advantage TSN offers compared to other real-time protocols lies in the fact that it is a standard. This increases interoperability and flexibility, as it allows OT and IT traffic to coexist on the same network seamlessly.

In the context of digital substations, TSN's role could be significant. Substations act as critical nodes within power grids, transforming voltage to suitable levels for efficient transmission and distribution of electricity. Furthermore, their digitalisation is a crucial element in energy transition. The integration of TSN into these substations could improve the manageability and reliability of power grids. The IEC technical report presented in [17] discusses design considerations for future deterministic networking technologies and how TSN can be applied in the context of digital substations. Digital substations handle various types of traffic, including trip messages, control data, and sample values. In this thesis, our objective is to conduct a thorough performance evaluation of the IEC 61850 communication protocol within a digital substation environment, comparing its performance when utilising either TSN or non-real-time Ethernet. The goal is to explore the potential benefits of TSN in digital substations by examining various feature configurations. To accomplish this, we will conduct several experiments in simulated networks to benchmark and analyse the real-time performance of IEC 61850 messages. The motivation gives rise to the following research questions:

> **RQ1:** What are the typical latency levels of IEC 61850 process bus protocol messages allocated in different TSN classes compared to a standard Ethernet network?
>
> **RQ2:** What is the observed jitter of IEC 61850 process bus protocol messages allocated in different TSN classes compared to a standard Ethernet network?
>
> **RQ3:** What is the effect on clock synchronisation by classifying gPTP traffic as ST?

In regards to RQ1, the IEC 61850 process bus carries critical messages from primary devices in the switchyard and IEDs. These messages must meet their deadlines to ensure the safety of both humans and equipment. Currently, the industry norm is to use standard Ethernet. However, in this thesis, we aim to explore how TSN can affect End-to-End delay and provide real-time guarantees for critical traffic.

RQ2 aims to analyse the jitter levels of the IEC 61850 process bus protocol messages. Since messages such as SV are sent periodically, reducing jitter can benefit applications dependent on this periodic information. Therefore, it is important to examine how different TSN classes affect the jitter levels of these messages and compare their performance to messages sent over standard Ethernet.

Finally, RQ3 aims to analyse how clock synchronisation is affected by classifying gPTP traffic as ST. For TSN networks it is imperative that all devices within the network are synchronised to ensure the arrival of all traffic. This is especially the case for traffic utilising TAS over several links. By treating gPTP traffic as ST we are not only scheduling it, but also protecting it. We want to investigate whether a TSN network can improve synchronisation accuracy or maintain synchronisation requirements compared to an Ethernet network. In both network types, all devices will be synchronised using gPTP.

To answer the proposed research questions we will incorporate the method described in the following section.

5 Method

To answer the proposed research questions of the thesis, we will use a modified version of the multimethodological research approach proposed by Nunamaker *et al.* [6]. Their proposed approach consists of a four-stage iterative process consisting of theory building, experimentation, observation, and systems development, see Figure 7. They argue that the proposed multimethodological approach is necessary for Information Systems (IS) research considering organisational acceptance and technological advancements. They further explain the four stages:

Theory building: Relevant research is gathered for the target domain, allowing generalisation of what has previously been done. Granting experiments, observation, and systems development to be performed.

Experimentation: Potential hypotheses and theories formulated during the theory building are developed and validated. Results from experiments are used to either expand on the developed theory or help further observation and systems development.

Observation: When a research domain is relatively unexplored case studies, survey studies, or field studies might need to be performed to develop hypotheses which can later be tested in experiments.

Systems development: The system is developed following the five steps: concept design, constructing the architecture of the system, prototyping, product development, and technology transfer.

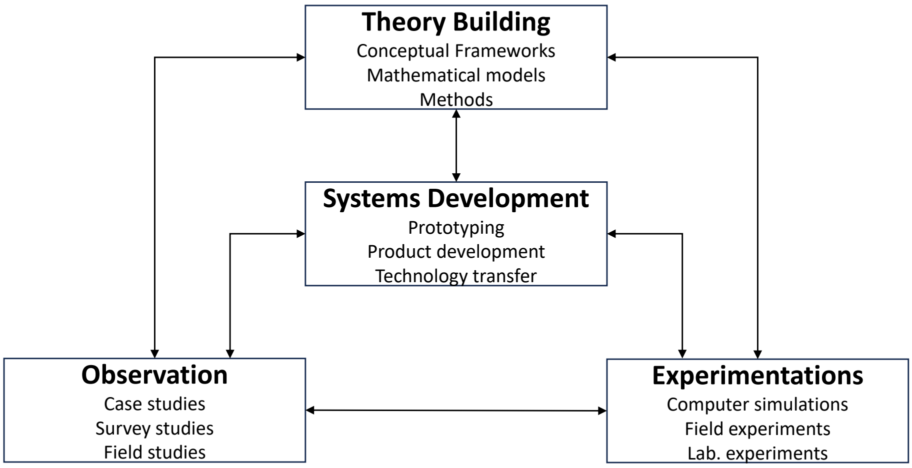

Figure 7: Multimethodological research approach for systems development

5.1 Applied Research Methodologies

The method proposed by Nunamaker *et al.* [6] serves as a general method for systems development, meaning the model has to be modified to able to fulfil the requirements for the specific application. For this thesis, a multimethodological approach, inspired by Nunamaker *et al.*, seen in Figure 8 is applied.

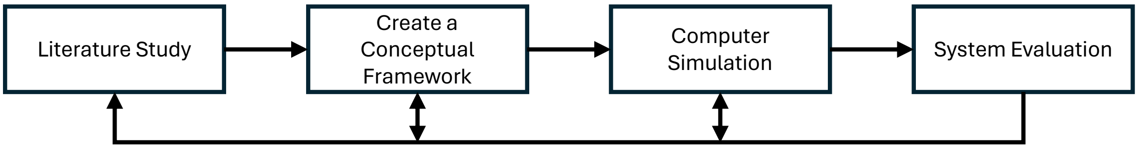

Figure 8: Flowchart of the applied research methods during the thesis

The thesis is initiated with a (I) literature study which aims to provide knowledge of the research field and research domain. The literature study provides the current state-of-the-art in the research field and finds related studies to the research topic. During the literature study, a (II) conceptual framework is formed. Here, the focus lies on understanding the process of system functionalities and procedures. It is also during this phase when studies are performed on relevant approaches for new ideas, which is a crucial step preceding experiments [6]. Experiments in this thesis are performed via (III) computer simulations. Simulations are beneficial in early-concept testing for experimentation with a high variety of parameters whilst keeping costs minimal. From the results of the computer simulations, an (IV) evaluation can be drawn. During this point, it is determined if the results satisfy the proposed research questions or if previous stages need to be revisited. The approach is designed to be iterative to combat any issues found during the process. During the thesis, the multimethodological approach will be iterated until an evaluation which addresses all the research questions is complete.

5.2 Tools

This section provides an overview of the tools utilised in this thesis.

5.2.1 OMNeT++

OMNeT++[2] is a simulation tool mainly utilised for the modelling of computer networks, either wired or wireless. It provides an open-source C++ object-oriented framework which is extensible and modular. This extensibility has resulted in many libraries that extend OMNeT++ such as INET, OverSim, and SimuLTE to name a few. Furthermore, OMNeT++ uses a descriptive modelling language called Network Description Language (NED) to define network topology and connections. In this thesis, we have used OMNeT++ 6.0.2 to create an Ethernet network topology of a digital substation process bus and to simulate traffic during different scenarios.

5.2.2 INET

INET[3] is an extensive model library for OMNeT++ that contains models for various network protocols, wireless and wired devices, and complex network simulations, supporting both research and educational purposes. In this thesis, INET 4.5.2 has been especially valuable for its Internet stack models, TSN models, and link layer protocols. It has significantly enhanced our ability to model a functional Ethernet and TSN network.

5.2.3 HERMES

HERMES is a heuristic scheduling algorithm for TSN developed at Mälardalen University by Bujosa *et al.* [41]. The scheduler is developed to arrive at a feasible schedule quickly and can be configured to have zero or relaxed reception jitter. In our thesis, we have used HERMES to arrive at a feasible schedule for our SV and gPTP traffic.

5.2.4 AVB Schedulability Analysis Tool

Ashjaei *et al.* [42] developed a schedulability analysis tool for AVB traffic. In their paper, the authors demonstrate that bandwidth over-reservation is necessary to provide real-time guarantees, and they propose a solution to achieve this reservation. In this thesis, we utilised this analysis tool to perform a worst-case delay analysis on GOOSE traffic which was classified as AVB traffic.

---

[2] https://OMNeT++pp.org/

[3] https://inet.omnetpp.org/

6 Ethical and Societal Considerations

Exploring the use of TSN in digital substations pose few ethical and societal issues due to the nature of this thesis being an early investigation for the potential integration of TSN into digital substations. However, electrical substations are considered critical infrastructure to society due to power losses potentially having major repercussions. Therefore safety and reliability must be of the highest concern when implementing new features into the digital substation. If the use of TSN would be considered as an alternative for future digital substation solutions, the connected TSN network would need to be tested rigorously in a worst-case scenario to ensure it can operate under all circumstances without endangering the safety of humans and equipment. To test a variety of parameters the thesis will only consist of simulations, meaning no testing will be done on physical equipment and therefore entailing that the safety of humans and equipment can be disregarded during this thesis.

7 Experimental Design and Implementation

The following sections detail the experimental setup and execution procedures of the simulations designed to address the proposed research questions. We aimed to evaluate three network scenarios: Ethernet without priorities, Ethernet with priorities, and TSN utilising TAS and CBS. Each scenario was further explored under two conditions: one simulating a low traffic load and the other a high traffic load. Table 5 presents descriptions and identifiers for these scenarios, which are used consistently throughout the remaining part of the report. The peak load for a link in a low-load scenario is around 10% utilisation. In contrast, during high traffic load scenarios, the average utilisation reaches approximately 90% on links carrying best-effort and MMS traffic. Furthermore, strict priority queuing ensures that traffic with higher priority is processed and transmitted before that of lower priority. Although this approach can reduce delays for higher priority traffic, lower priority traffic can be starved, where it may receive little or no service during heavy load periods. Traffic loads and priorities are discussed in more detail in Section 7.2.1 and 7.2.2.

Scenario ID	**Description**
E-L	Standard Ethernet with low traffic load
E-H	Standard Ethernet with high traffic load
E-PQ-L	Standard Ethernet with strict priority queueing and low traffic load
E-PQ-H	Standard Ethernet with strict priority queueing and high traffic load
TSN-L	TSN with TAS, CBS, and low traffic load
TSN-H	TSN with TAS, CBS, and high traffic load

Table 5: Summary of simulation scenarios

Additionally, we have created a fork of the current INET branch on our git repository[4] that allows the user to perform further simulations. The main contributions of the fork are the implemented IEC 61850 protocols and TSN compatibility with layer-2 traffic, which is explained in more detail in 7.5.1. The fork also contains Python scripts for quick statistical extraction and visualisation.

7.1 Network Topology Design

In designing the network topology, we incorporated ideas from prior studies [9], [38] to emulate an IEC 61850 process bus topology commonly used in the field. Additionally, we considered the demands of Industry 4.0, which includes a rise in IoT devices within industrial networks. IT and OT-traffic are expected to be able to share the same network without disruptions or performance loss for critical traffic [43]. To simulate this, we introduced several low-priority nodes into our model that generate additional best-effort traffic, potentially causing interference. The resulting network topology is illustrated in Figure 9. This figure shows three bays that contribute similarly to network traffic. Inside each bay, SV traffic is generated from the MUs and multicasted to the IEDs in their respective bay, similarly the IEDs generate GOOSE traffic destined for the MUs. Additionally, MMS traffic is exchanged between two IEDs within each bay. The internet hosts and a log server simulate traffic flowing to and from the higher layers of a digital substation via the process bus, as shown in Figure 2. Tables 6 and 7 provide detailed breakdowns of all simulated traffic flows during low and high traffic load scenarios.

---

[4] https://github.com/s4mwag/thesis_tsn_substation

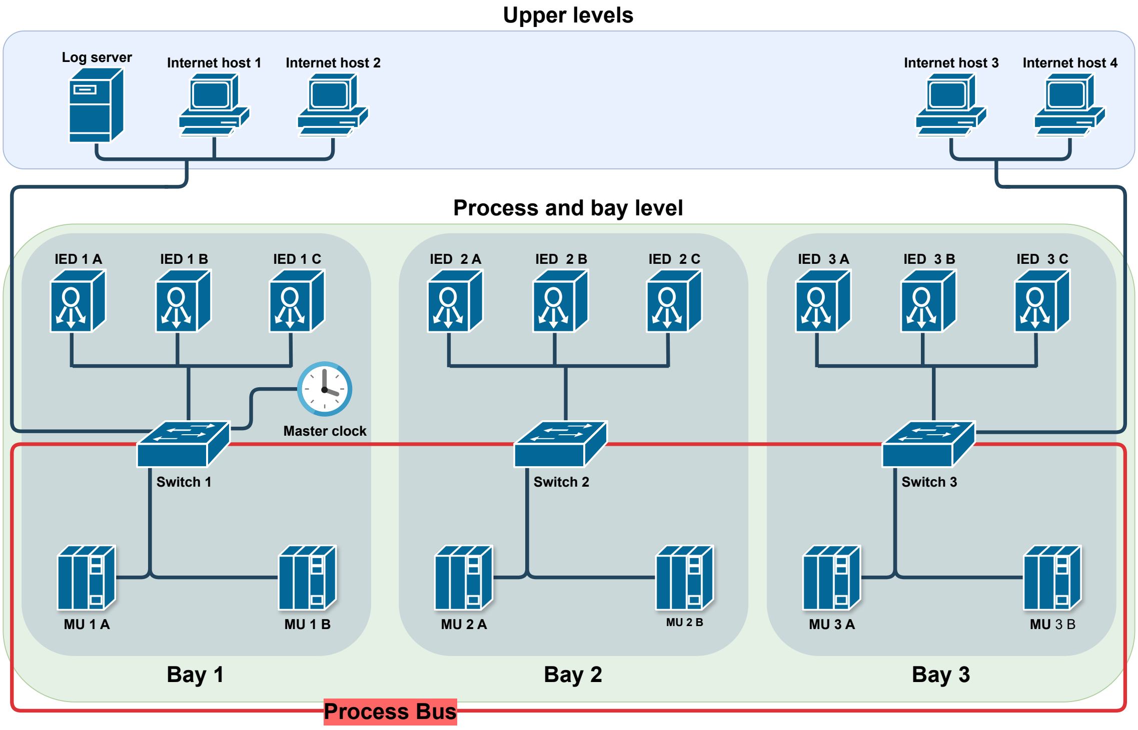

Figure 9: Network topology used during simulations

7.2 Traffic Design

To ensure communication resilience within SAS the IEC 61850 protocol [17] employs various algorithms within protocols to ensure that high-priority messages are delivered within their respective deadlines. Traffic types which send data periodically, such as SV messages, achieve resilience by sampling and sending data at a high frequency. However, the GOOSE protocol sends messages at a sporadic rate meaning that the sending interval of GOOSE messages cannot be pre-defined. To ensure that no GOOSE messages are lost in transmission, the IEC 61850 standard [17, p. 52] defines the algorithm for how GOOSE messages shall be sent in bursts, see Figure 10.

When the system is in a steady state, i.e. no events are triggered, periodic heartbeat messages are sent with a period of *T0*. When an event occurs, a burst of five messages containing the same data will be sent to ensure the arrival of the information. The first copy will be sent with a delay of *T1* and the second copy will be sent with a delay of *T1*+*T1* from the first message. The third copy is sent with a delay of *T1*+*T1*+*T2* from the first message and the fourth copy is sent with a delay of *T1*+*T1*+*T2*+*T3* from the first message. If a new event is triggered before all copies of the previous event have been transmitted, the new event will take precedence and the remaining copies of the previous event will not be sent. When an event occurs, it will temporarily pause the transmission of the heartbeat messages until the transmission period reaches a steady state, as depicted in Figure 10. The heartbeat was set to a period of 500 ms and *T1* was set to 1 ms [5]. *T2* and *T3* are application dependent, as stated in the IEC 61850 standard [17, p. 52]. For our application, *T2* was set to 100 ms and *T3* to 250 ms.

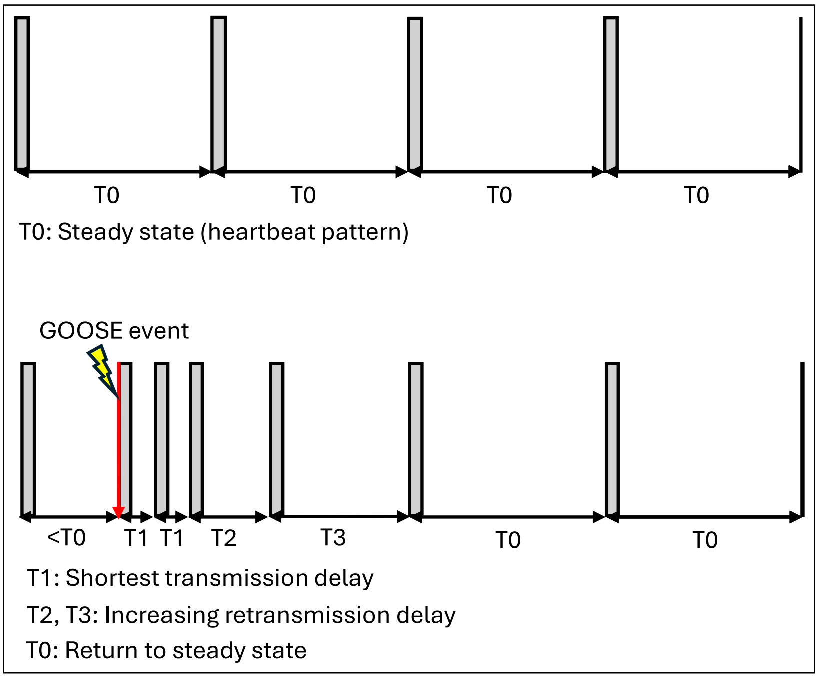

Figure 10: GOOSE transmission and re-transmission pattern

7.2.1 Traffic Flows

Table 6 and 7 provide detailed descriptions of all traffic flows for the low and high traffic load scenarios, respectively. Differences between these scenarios are highlighted in red in Table 7. Generally, for the high traffic load scenario, the volume of MMS, logging, and IT traffic is increased to nearly fully utilise the bandwidth capacity of the links they traverse.

The traffic flows are designed with a digital substation in mind, as mentioned in Section 7.1. Hence, in each respective bay, SV traffic is flowing from the MUs, multicasted to every IED in that bay at a rate of 4 kHz. GOOSE 1A and GOOSE 1B traffic are sent from two IEDs and multicasted to both MUs in each bay. Between IED A and IED B in each bay, MMS traffic is sent. GPTP traffic is sent in the hierarchical fashion according to its protocol and is dependent on gPTP message type, the master clock node is the gPTPclock node. Logging messages are sent from every device and flow via the fastest route to the log server. Internet hosts exchange messages and the routes have been fixed to simulate a load-balancing behaviour so that not all IT traffic traverses the same links. In the low traffic load scenarios, MMS and IT traffic flows are roughly around 1 Mbit/s per flow while in the high-load scenarios, the links are loaded to utilise between 80 - 100 Mbit/s of the bandwidth.

Flow ID	Description	Source nodes	Destination nodes	Transmission type	Frame size	Traffic rate
1	SV from MUs Bay 1	MU_1_A, MU_1_B	IED_1_A+B+C	Multicast	140 bytes	250 µs
2	SV from MUs Bay 2	MU_2_A, MU_2_B	IED_2_A+B+C	Multicast	140 bytes	250 µs
3	SV from MUs Bay 3	MU_3_A, MU_3_B	IED_3_A+B+C	Multicast	140 bytes	250 µs
4	GOOSE 1A Trip Bay 1	IED_1_A, IED_1_B	MU_1_A+B	Multicast	160 bytes	20 events · sim(s)
5	GOOSE 1A Trip Bay 2	IED_2_A, IED_2_B	MU_2_A+B	Multicast	160 bytes	20 events · sim(s)
6	GOOSE 1A Trip Bay 3	IED_3_A, IED_3_B	MU_3_A+B	Multicast	160 bytes	20 events · sim(s)
7	GOOSE 1B Trip Bay 1	IED_1_B, IED_1_C	MU_1_A+B	Multicast	160 bytes	20 events · sim(s)
8	GOOSE 1B Trip Bay 2	IED_2_B, IED_2_C	MU_2_A+B	Multicast	160 bytes	20 events · sim(s)
9	GOOSE 1B Trip Bay 3	IED_3_B, IED_3_C	MU_3_A+B	Multicast	160 bytes	20 events · sim(s)
10	MMS traffic Bay 1	IED_1_A	IED_1_B	Unicast	150 bytes	833 events · sim(s)
11	MMS traffic Bay 2	IED_2_A	IED_2_B	Unicast	150 bytes	833 events · sim(s)
12	MMS traffic Bay 3	IED_3_A	IED_3_B	Unicast	150 bytes	833 events · sim(s)
13	gPTP PdelayRequest	IEDs, MUs, and switches	Master port neighbor	Unicast	80 bytes	1 s
14	gPTP PdelayResp	gPTPclock and switches	Slave port neighbors	Unicast	80 bytes	∼1 s
15	gPTP DelayFollowUp	gPTPclock and switches	Slave port neighbors	Unicast	80 bytes	∼1 s
16	gPTP Sync	gPTPclock	Slave nodes	Multicast	72 bytes	31.25 ms
17	gPTP FollowUp	gPTPclock	Slave nodes	Multicast	102 bytes	∼31.25 ms
18	Log messages	IEDs and MUs	Log_server	Unicast	150 bytes	5 events · sim(s)
19	IT traffic via switch 2	internet_host_2	internet_host_4	Unicast	1500 bytes	83 events · sim(s)
20	IT traffic via switch 2	internet_host_4	internet_host_2	Unicast	1500 bytes	83 events · sim(s)
21	IT between switch 1 and 3	internet_host_1	internet_host_3	Unicast	1500 bytes	83 events · sim(s)
22	IT between switch 1 and 3	internet_host_3	internet_host_1	Unicast	1500 bytes	83 events · sim(s)

Table 6: Traffic flow description for low-load scenarios

Flow ID	Description	Source nodes	Destination nodes	Transmission type	Frame size	Traffic rate
1	SV from MUs Bay 1	MU_1_A, MU_1_B	IED_1_A+B+C	Multicast	140 bytes	250 µs
2	SV from MUs Bay 2	MU_2_A, MU_2_B	IED_2_A+B+C	Multicast	140 bytes	250 µs
3	SV from MUs Bay 3	MU_3_A, MU_3_B	IED_3_A+B+C	Multicast	140 bytes	250 µs
4	GOOSE 1A Trip Bay 1	IED_1_A, IED_1_B	MU_1_A+B	Multicast	160 bytes	20 events · sim(s)
5	GOOSE 1A Trip Bay 2	IED_2_A, IED_2_B	MU_2_A+B	Multicast	160 bytes	20 events · sim(s)
6	GOOSE 1A Trip Bay 3	IED_3_A, IED_3_B	MU_3_A+B	Multicast	160 bytes	20 events · sim(s)
7	GOOSE 1B Trip Bay 1	IED_1_B, IED_1_C	MU_1_A+B	Multicast	160 bytes	20 events · sim(s)
8	GOOSE 1B Trip Bay 2	IED_2_B, IED_2_C	MU_2_A+B	Multicast	160 bytes	20 events · sim(s)
9	GOOSE 1B Trip Bay 3	IED_3_B, IED_3_C	MU_3_A+B	Multicast	160 bytes	20 events · sim(s)
10	MMS traffic Bay 1	IED_1_A	IED_1_B	Unicast	150 bytes	83,330 events · sim(s)
11	MMS traffic Bay 2	IED_2_A	IED_2_B	Unicast	150 bytes	83,330 events · sim(s)
12	MMS traffic Bay 3	IED_3_A	IED_3_B	Unicast	150 bytes	83,330 events · sim(s)
13	gPTP PdelayRequest	IEDs, MUs, and switches	Master port neighbor	Unicast	80 bytes	1 s
14	gPTP PdelayResp	gPTPclock and switches	Slave port neighbors	Unicast	80 bytes	∼1 s
15	gPTP DelayFollowUp	gPTPclock and switches	Slave port neighbors	Unicast	80 bytes	∼1 s
16	gPTP Sync	gPTPclock	Slave nodes	Multicast	72 bytes	31.25 ms
17	gPTP FollowUp	gPTPclock	Slave nodes	Multicast	102 bytes	∼31.25 ms
18	Log messages	IEDs and MUs	Log_server	Unicast	150 bytes	10,000 events · sim(s)
19	IT traffic via switch 2	internet_host_2	internet_host_4	Unicast	1500 bytes	5050 events · sim(s)
20	IT traffic via switch 2	internet_host_4	internet_host_2	Unicast	1500 bytes	5050 events · sim(s)
21	IT between switch 1 and 3	internet_host_1	internet_host_3	Unicast	1500 bytes	5050 events · sim(s)
22	IT between switch 1 and 3	internet_host_3	internet_host_1	Unicast	1500 bytes	5050 events · sim(s)

Table 7: Traffic flow description for high-load scenarios

7.2.2 Priority Mapping

As previously mentioned in Section 3.4 and outlined in Table 3, Docquier *et al.*[5] recommended specific IEC 61850 to TSN traffic mappings. Our approach, detailed in Table 8, follows a similar structure but includes modifications to better accommodate gPTP synchronisation messages. We prioritise synchronisation traffic above MMS, logging, and IT traffic due to its critical nature. Consequently, scheduled gPTP traffic is assigned to priority 4, while its response traffic is placed in priority 3. MMS traffic, although less critical than synchronisation traffic, is still prioritised above other best-effort traffic. In line with the IEC 61850 standard [44, pp. 76-78], which allows MMS deadlines ranging from 1000 ms to 100 ms depending on the traffic type, we have opted for the stricter deadline of 100 ms to ensure robust performance.

Traffic type	TSN Traffic class	Priority	Deadline	Protocol	Traffic pattern
SV	ST	7	3 ms	Ethernet (88-BA)	Periodic
GOOSE 1A	Class A	6	3 ms	Ethernet (88-B8)	Sporadic
GOOSE 1B	Class B	5	20 ms	Ethernet (88-B9)	Sporadic
gPTP	ST	4	-	gPTP	Periodic
gPTP response traffic	Best-effort	3	-	gPTP	Aperiodic
MMS	Best-effort	2	100 ms	TCP/IP	Aperiodic
Log traffic, internet traffic	Best-effort	0	-	TCP/IP	Aperiodic

Table 8: Traffic type characteristics

7.3 Clock Synchronisation Design

Design choices for synchronisation in our network follow the guidelines suggested by Gutierrez *et al.* [45] for industrial automation networks. However, while the paper advocates for the one-step mode of gPTP, OMNeT++, supports only two-step synchronisation[5]. This means that for each *Sync* and *PdelayResp* message in our setup, there is an accompanying *FollowUp* message, unlike in the one-step mode where all synchronisation data is incorporated into a single message. Following Gutierrez *et al.*, we have implemented a synchronisation period of 31.25 ms and a *PdelayRequest* period of 1 second. This synchronisation traffic is regular and thus can be scheduled using TAS, while response traffic, which is event-based, is categorised as aperiodic.

The clock synchronisation performance will be measured by comparing the current time of the master clock node in Figure 9 to all the synchronised nodes in the topology. The master clock is ideal, i.e. it keeps the same time as the simulated time. The synchronised nodes are contained in the process and bay level and will be experiencing random clock drift configured with a standard INET configuration. Specific configuration can be viewed in the git repository for the thesis, which can be found in Section 7.

7.4 Design Constraints

In the development of a TSN environment, the type of interface used on end nodes can affect their ability to support real-time data processing. Interfaces designed specifically for TSN can offer better real-time performance than those using standard Ethernet, as they may include traffic shapers such as TAS and CBS. However, for this thesis, we have modelled all end nodes as non-real-time. This means that even though they are synchronised, they do not use any Real-Time Operating System (RTOS) to manage application priorities, nor do they have any form of priority queuing for traffic at the node level. Instead, traffic shaping and priority queuing are handled solely by the network switches in our simulations. Furthermore, TSN supports frame preemption which can reduce End-to-End delay for higher priority traffic. However, in our simulations, we have not implemented this.

7.5 OMNeT++ Implementation and Customisation

The network topology and traffic flows discussed in Section 7.1 and 7.2 were implemented in the latest versions of OMNeT++ and INET. There are several reasons for this. First, TSN features are continuously being developed and maintained on OMNeT++ INET's git master branch while NeSTiNg has not been updated in three years. Second, while some previous works exist that have implemented IEC 61850 protocols in OMNeT++ and INET, such as [1], the models were created for older versions of OMNeT++. However, updating these models to be compatible with current versions of OMNeT++ is complex, especially when the original code was designed for much earlier versions. As a result, in this thesis, we have decided to implement the IEC 61850 protocols as required for our simulation in the newer environment.

7.5.1 Traffic Implementation

The *EthernetSourceApp* in INET was utilised to generate SV and GOOSE traffic. However, when enabling outgoing streams on TSN devices, INET introduces a bridging layer to manage the identification, encoding, and decoding of packet streams. Consequently, the specific streams correctly receive the Priority Code

---

[5] https://inet.OMNeT++pp.org/docs/showcases/tsn/timesynchronization/gptp/doc/index.html

Point (PCP)-field when the Ethernet module applies the 802.1Q-tag, as illustrated in Figure 11. However, in the current version of INET, this bridging layer lacks compatibility with layer-2 traffic. As a result, we needed to modify INET's message dispatcher and packet multiplexer module. This modification allowed us to bind Ethernet messages to the Ethernet module and facilitate the passage of Ethernet messages through the bridging layer. SV traffic was easily generated by applying a production interval of 250 µs for messages. However, generating GOOSE messages posed more of a challenge. Since INET lacks a traffic generation application that simulates the behaviour of GOOSE messages, we developed and implemented our own GOOSE application for INET. The behaviour of GOOSE is described in Section 7.2 and depicted in Figure 10. Now, GOOSE can be incorporated into the simulation configuration file by setting the parameter *source.useGoose* to true in the *EthernetSourceApp* and specifying the number of GOOSE events per second using the parameter *source.numberOfGooseEventsPerSec*.

Implementing other types of traffic was straightforward. As MMS, log, and IT-traffic are event-based TCP traffic types, we utilised INET's *TcpSessionApp*. This involved automatically generating seed-dependent text files, which could then be passed to the application's *sendScript* parameter. Additionally, adjusting the traffic load from high to low and vice versa was done depending on the simulated scenario. Further details regarding the simulation execution are provided in Section 7.7.

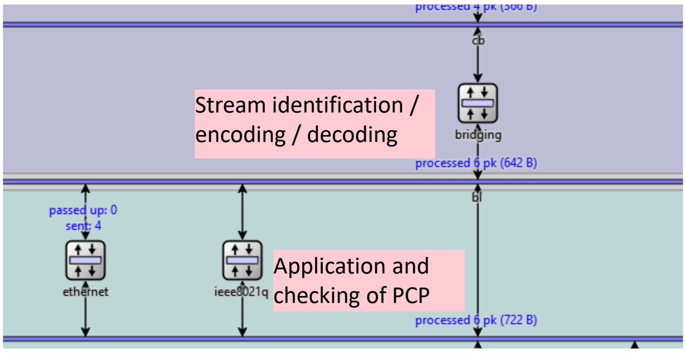

Figure 11: PCP handling inside a TSN device in OMNeT++

7.6 Scheduling and Analysis

As described in Section 2.5.3, for TAS to operate the states of the different gates a GCL has to be configured. The GCL for our application is created using the scheduler HERMES for TSN proposed by Bujosa *et al.* [41] and will schedule all ST in the network. For our application, SV and gPTP messages will be scheduled by HERMES. To successfully schedule the traffic, HERMES requires the following information; the different links the individual message traverses, the sending frequency of the message, and the size of the message. HERMES will schedule the messages towards the end of the message period, which creates a higher end-to-end delay for the message but still within its deadline. However, it is important to note that for real-time traffic it is essential to guarantee that traffic can meet its deadline rather than minimising End-to-End delay. The final output from HERMES is a schedule detailing when each message is allocated a time slot on a link along its path to the destination.

To apply the schedule created by HERMES into OMNeT++, a program was created to convert the HERMES output to a GCL compatible with OMNeT++. The approach used will open the gates for the ST as assigned by HERMES and close all other gates. When none of the ST is scheduled to be sent, all other gates will be opened to allow best-effort and event-based messages to be sent. As seen in Table 8, GOOSE 1A and GOOSE 1B were allocated the highest priority of the non-ST. To ensure that the best-effort traffic does not suffer from starvation, CBS was applied to the GOOSE 1A and GOOSE 1B queues. To determine the correct idleSlope of the credits for CBS, the link utilisation for the links containing GOOSE messages had to be calculated. When the link utilisation was calculated, the idleSlope for Class A and Class B was assigned a percentage of the maximum link bandwidth equal to their link utilisation, see Table 9 for the resulting idleSlopes applied to the TSN network.

	Class A (Mbps)	Class B (Mbps)
idleSlope	49.6	49.6

Table 9: IdleSlopes for Class A and Class B for switch interfaces connecting to MUs. These values are replicated through all bays

Once the idleSlopes were configured for the CBS queues, it was necessary to determine if a scenario exists where the worst-case response time for the GOOSE messages exceeds their deadlines. The AVB-analysis tool developed by Ashjaei *et al.* [42] was used to compute the worst-case response time for the GOOSE messages. It should stated that the AVB-analysis tool is designed to be pessimistic when computing the worst-case response time for Class B traffic, i.e. the computed worst-case response time for the Class B messages will be higher than the actual worst-case case response time. To compute the worst-case response time, the tool requires a formatted version of the generated schedule by HERMES which states the GOOSE messages' characteristics. It is also necessary to define the link utilisation of the links which are using CBS. After the execution of the AVB-analysis tool, it outputs the worst-case execution time for the GOOSE messages. If the computed worst-case response times are within the deadline for GOOSE messages we know that TSN with CBS can guarantee the arrival of GOOSE messages within its deadline.

7.7 Simulation Execution

To ensure the integrity of results, a sufficient number of simulations should be run. However, in this thesis, simulations for all scenarios were rerun in a series of 50 iterations. This number of iterations was the maximum possible due to hardware limitations. Because of the number of events and the length of simulations, the result files were upwards of 10 GB in size, which severely restricted the number of simulations that could be stored on the available hard drives. This resulted in a total of 300 simulation runs. The simulations were identical between iterations except when the event-based messages were triggered. The events for the event-based messages were randomised using a seed, for each iteration the seed was updated. Using a seed to generate the randomly triggered events allows the six scenarios to be exposed to the same randomly triggered events. This entails that all six scenarios have been exposed identically, all to ensure each scenario is analysed equally.

The simulations were initialised and controlled by a shell script. The script iterates the number of times specified by the user, in our case 50 iterations. Each iteration was initialised by executing a Python program which randomly generated the time stamps and packet size for all MMS traffic for that iteration and saved it to a text file. Following the generation of the MMS traffic, the simulation was initialised with the seed passed as an argument. The seed for the simulation was used to randomly generate the events for the GOOSE messages. The same seed was used for the generation of the MMS traffic and the GOOSE traffic, the seed was equal to the current iteration of the simulation.

8 Results

This section presents results gathered from the six different simulation scenarios summarised in Table 5 and detailed in Section 7. In total, there were 50 simulations for each scenario. Section 8.1 addresses End-to-End delay measurements, Section 8.2 shows jitter results, and Section 8.3 covers the clock synchronisation results.

8.1 End-to-End Delay

This subsection presents the resulting End-to-End delay for SV, GOOSE, and MMS messages. Statistics from each simulation scenario will be displayed so that they can be analysed and compared to each other.

8.1.1 SV Messages

Figure 12 displays a box plot of the End-to-End delay for SV messages. As discussed in Section 7.6, the HERMES scheduling strategy places messages towards the end of the period, and as a result, reduces jitter. This approach leads to increased End-to-End delays for SV messages and ST overall for TSN scenarios in comparison to non-TSN scenarios, as shown in the figure. However, the TSN scenarios exhibit fewer outliers, indicating more consistent message timing and lower jitter, which we will explore in more detail in Section 8.2. The findings are summarised in Table 10.

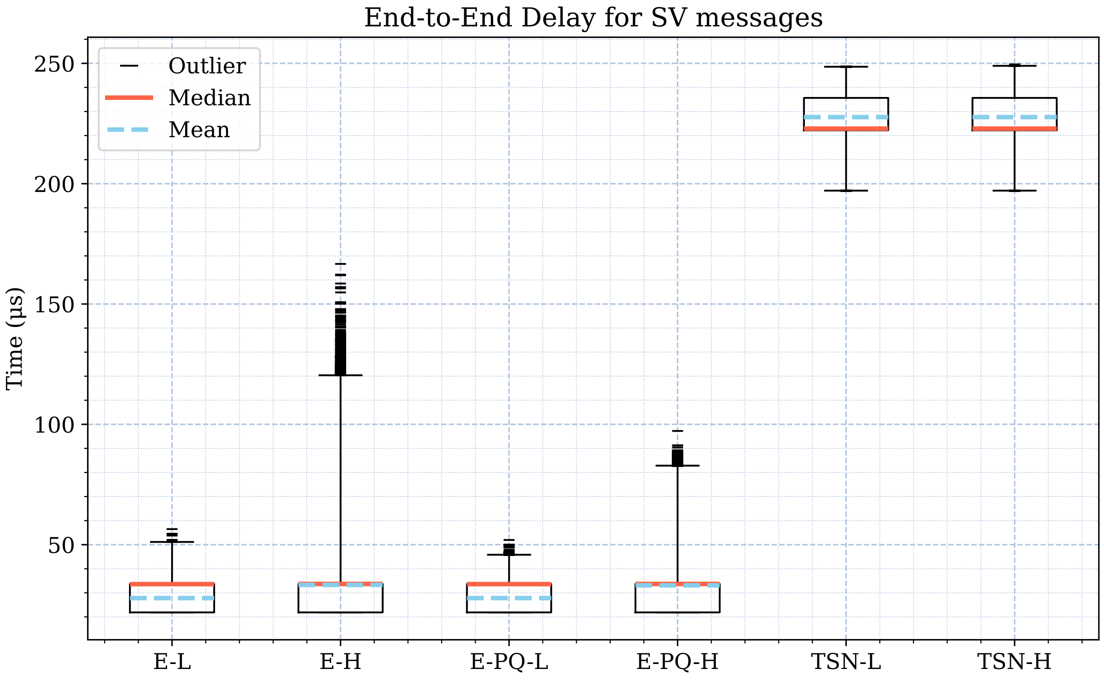

Figure 12: End-to-End delay for SV messages

Scenario	**Min.** (µs)	**Max.** (µs)	**Mean** (µs)	**SD** (µs)
E-L	22	56	28	6
E-H	22	167	33	13
E-PQ-L	22	52	28	6
E-PQ-H	22	97	33	13
TSN-L	197	249	228	13
TSN-H	197	249	228	13

Table 10: Summary of End-to-End delay for SV messages

8.1.2 GOOSE Messages

Table 11 shows the calculated worst-case response time from the AVB-analysis tool developed by Ashjaei *et al.* [42]. The results show that by classifying GOOSE traffic as AVB, TSN together with the CBS can guarantee the arrival of GOOSE messages within their 3 ms and 20 ms deadlines, as defined by IEC 61850 [44, pp. 76-78].

Message type	**AVB Class**	**Worst case response time** (ms)
GOOSE 1A	Class A	1.3378
GOOSE 1B	Class B	1.6005

Table 11: Schedulability analysis of GOOSE traffic

Figure 13 and Table 12 summarise the End-to-End delay results for GOOSE 1A messages which have the second highest priority after SV messages. It can be observed that priority queuing in the Ethernet scenarios will result in lower End-To-End delay if the traffic load is low. However, in the high traffic load scenarios, the End-To-End delay is decreased for TSN. The TSN scenarios display a slightly higher mean delay than the Ethernet scenarios.

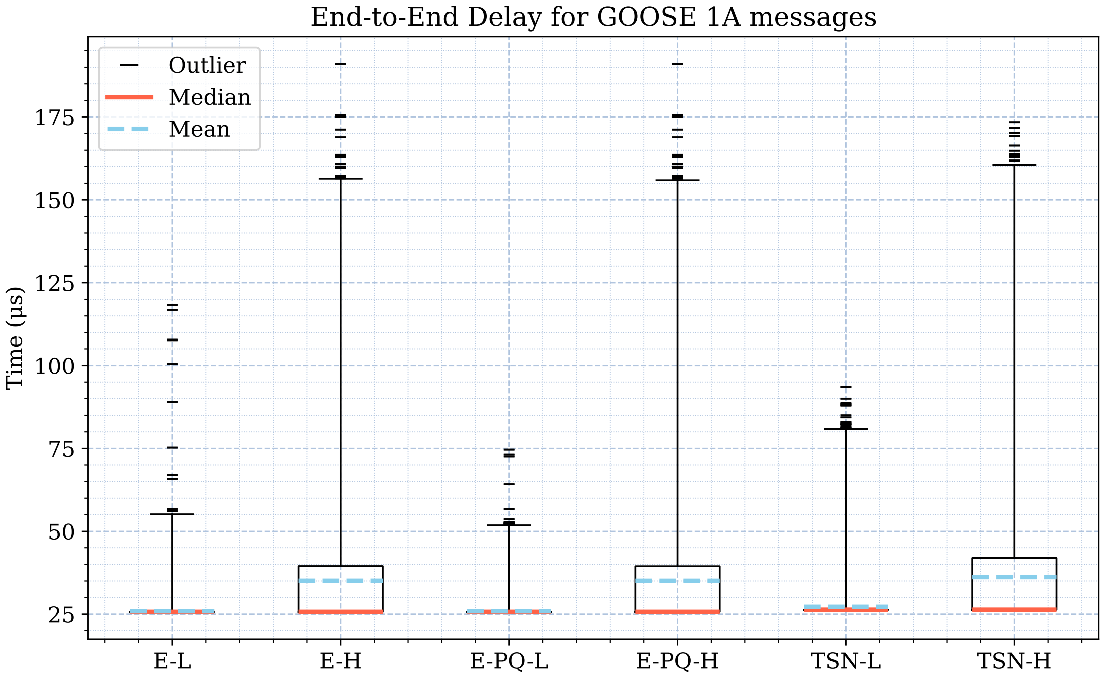

Figure 13: End-to-End delay for GOOSE 1A messages

Scenario	**Min.** (µs)	**Max.** (µs)	**Mean** (µs)	**SD** (µs)
E-L	26	118	26	1
E-H	26	191	35	16
E-PQ-L	26	75	26	1
E-PQ-H	26	191	35	16
TSN-L	26	96	27	3
TSN-H	26	173	36	16

Table 12: Summary of End-to-End delay for GOOSE 1A messages

Figure 14 and Table 13 summarise the End-to-End delay results for GOOSE 1B messages which have the third highest priority below SV and GOOSE 1A messages. It can be observed that except for small differences in the standard deviation, every scenario is almost identical between high and low traffic load. Furthermore, TSN only experienced a negligible increase in mean End-to-End delay compared to the Ethernet scenarios.

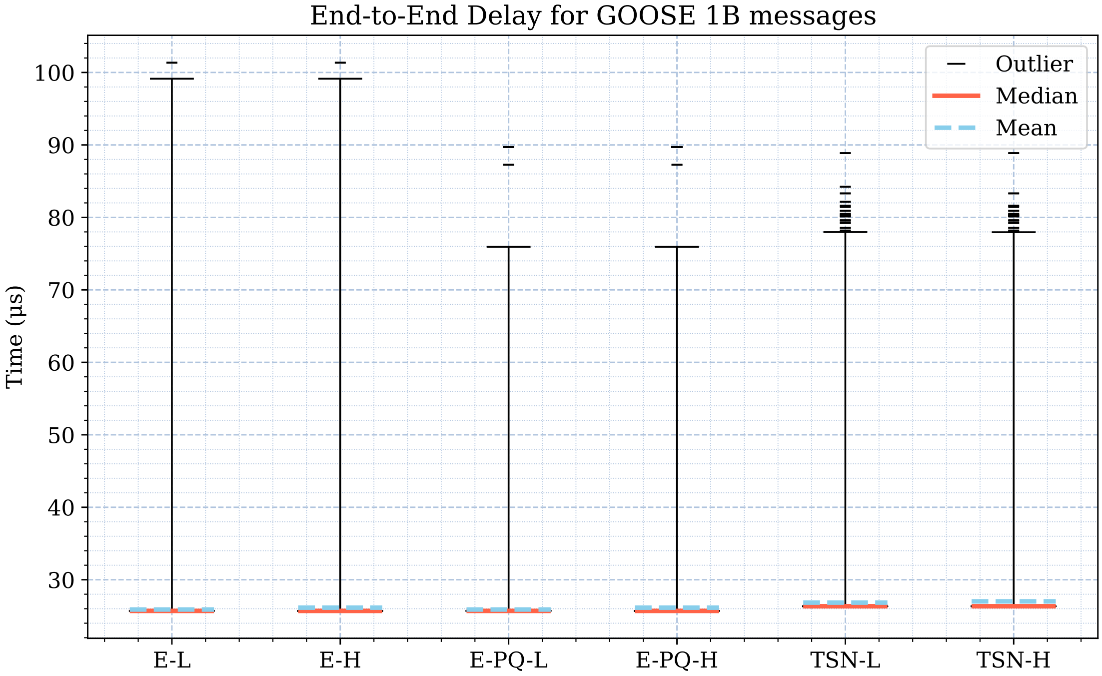

Figure 14: End-to-End delay for GOOSE 1B messages

Scenario	**Min.** (µs)	**Max.** (µs)	**Mean** (µs)	**SD** (µs)
E-L	26	101	26	1
E-H	26	101	26	2
E-PQ-L	26	90	26	1
E-PQ-H	26	90	26	2
TSN-L	26	89	27	3
TSN-H	26	89	27	3

Table 13: Summary of End-to-End delay for GOOSE 1B messages

8.1.3 MMS Messages

Figure 15 and Table 14 summarises the End-to-End delay for MMS messages. MMS is one of the traffic types that is increased during the high traffic load scenarios. It is evident that the End-to-End delay for MMS messages is increased for all scenarios during the high traffic load and that it may be increased due to the MMS messages being of lower priority than SV and GOOSE. The number of outliers suggests that the jitter would be high, which is confirmed by the higher standard deviation. However, since MMS messages fall in a lower priority queue and has a higher deadline than higher priority traffic it is acceptable. MMS messages during all scenarios meet the deadline of 100 ms.

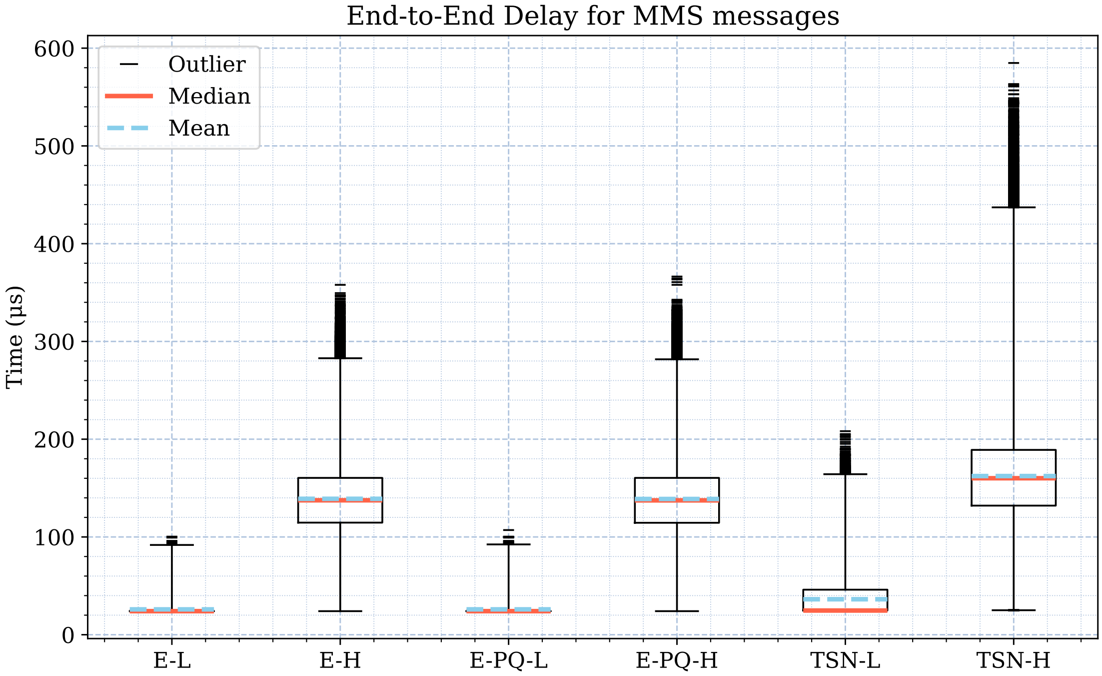

Figure 15: End-to-End delay for MMS messages

Scenario	**Min.** (µs)	**Max.** (µs)	**Mean** (µs)	**SD** (µs)
E-L	24	100	26	6
E-H	24	358	139	30
E-PQ-L	24	107	26	6
E-PQ-H	24	366	139	30
TSN-L	25	208	36	18
TSN-H	25	585	162	41

Table 14: Summary of End-to-End delay for MMS messages

8.2 SV Jitter

Figure 16 to 18 show the jitter of SV messages. The jitter is defined as the difference between packet delay of successive packets (instantaneous jitter). The figures only show messages from two MUs to one IED since the behaviour is representative of all SV traffic and creates a more distinct plot. The graphs suggest that Ethernet without priority queuing produces a higher maximum jitter than Ethernet with priority. It can also be seen that for TSN, the variability in delay is less than that of the Ethernet scenarios.

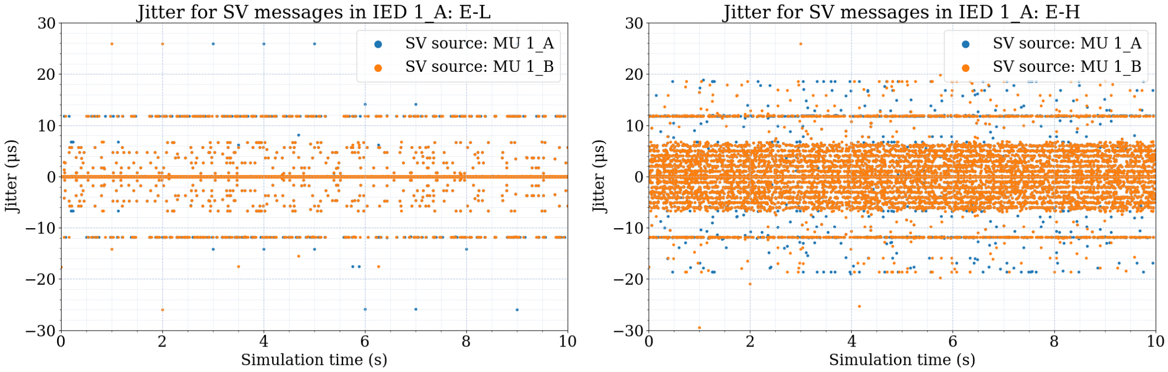

Figure 16: Jitter for SV messages in scenario E-L and E-H

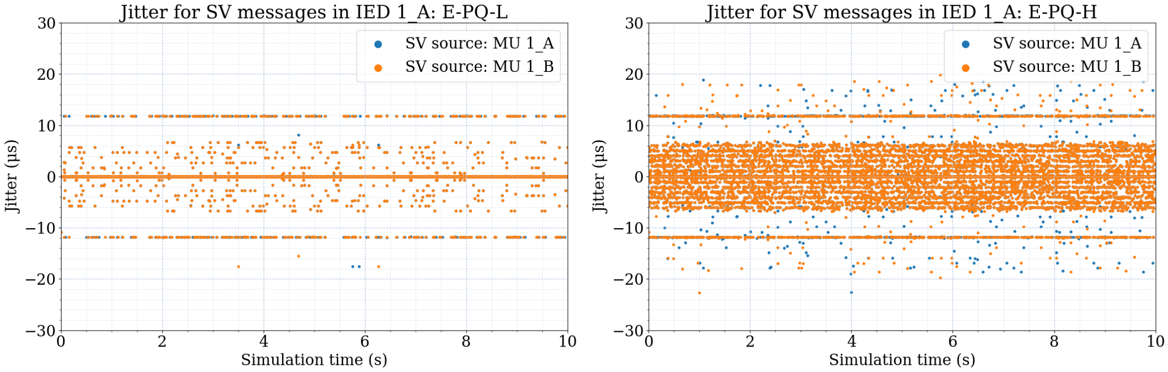

Figure 17: Jitter for SV messages in scenario E-PQ-L and E-PQ-H

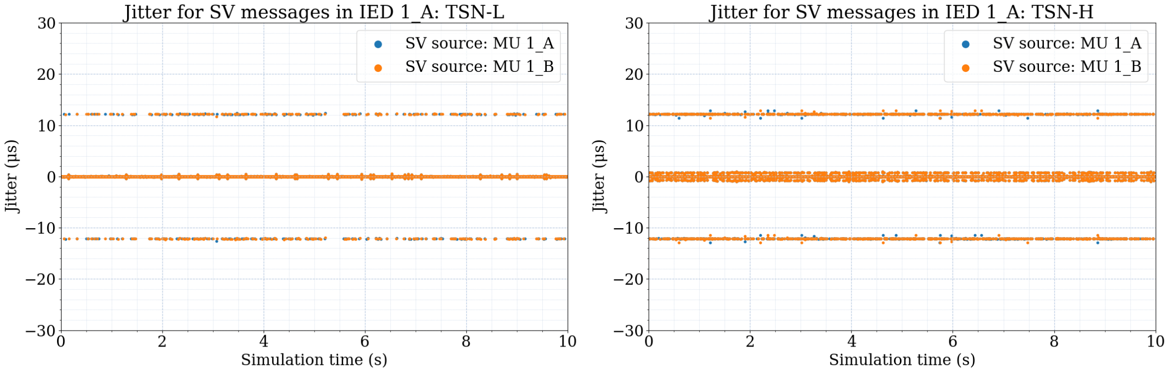

Figure 18: Jitter for SV messages in scenario TSN-L and TSN-H

Figure 19 presents the instantaneous jitter statistics for all SV traffic, which is the highest priority traffic. The graph serves as a complementary overview to Figure 16 to 18. The data indicates that jitter escalates for all configurations when the traffic load is increased. However, TSN displays less variability between the low and high traffic load scenarios compared to the Ethernet scenarios.

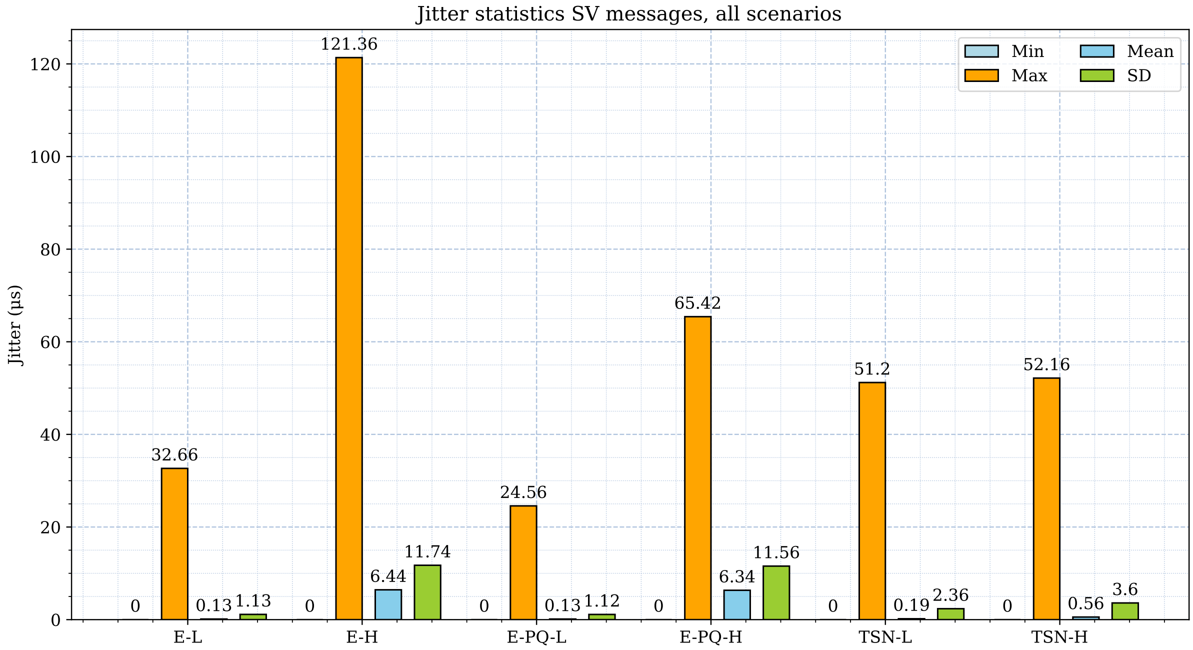

Figure 19: Jitter statistics SV messages, all scenarios

8.3 Clock Synchronisation

This results section explores the synchronisation errors within the different network scenarios. In the Ethernet scenarios, gPTP messages are handled both with and without priority, whereas in the TSN scenarios, periodic gPTP messages are treated as ST. We define the maximum synchronisation error as the largest time difference between any time-aware system and the master clock. The mean synchronisation error represents the average synchronisation error across all time-aware systems when compared to the master clock node. The master clock node, which, as established in Section 7.3, functions as an ideal clock.

Table 15 presents the gPTP clock synchronisation error results for all scenarios. In the Ethernet scenarios, we observed a smaller synchronisation error when there was a higher traffic load compared to scenarios with lower traffic. Conversely, the TSN scenarios showed the same error in both high and low traffic conditions but had the highest maximum and mean errors among all the scenarios.

Scenario	**Max. error** (ns)	**Mean error** (ns)
E-L	109	15
E-H	100	13
E-PQ-L	132	15
E-PQ-H	114	14
TSN-L	301	84
TSN-H	301	84

Table 15: Summary of clock synchronisation error in time-aware hosts

9 Discussion

In the following section, we will analyse and interpret the results in relation to our research questions, limitations, and implications for future studies. Furthermore, we will provide suggestions for further research. This analysis will encompass our three research questions and performance metrics: End-to-End delay, jitter, and clock synchronisation, each discussed in the following subsections.

9.1 End-to-End Delay

During the simulations of all scenarios, the End-to-End delay was recorded for SV, GOOSE, and MMS messages. In general, our results show that the TSN scenarios will produce a higher End-to-End delay. This is most evident when analysing the End-to-End delay for SV messages during TSN scenarios where the minimum End-to-End delay is higher than the maximum End-to-End delay for the non-TSN scenarios. The reason for this behaviour can be found in the TSN scheduler used for scheduling ST (SV and gPTP messages). HERMES schedules ST towards the end of the period to find a feasible schedule where all traffic is ensured to meet its deadlines at the expense of a higher End-to-End delay. The purpose of the HERMES is not to minimise End-to-End delay, but rather ensure all traffic meets all deadlines by finding a feasible schedule. If End-to-End delay needs to be reduced for ST, then an alternative scheduler would be required to ensure ST is sent as close to the start of the period as possible.

When analysing the End-to-End delay for GOOSE messages all scenarios achieve the same minimum End-to-End delay, and the mean End-to-End delay is similar for all low traffic load and high traffic load scenarios respectively. There are however differences when analysing and comparing the maximum End-to-End delay of TSN and non-TSN scenarios, see Table 12. During high-load scenarios, TSN with CBS manage to reduce the maximum End-to-End delay for GOOSE traffic. These findings showcase the effectiveness of CBS during traffic congestion. If achieving lower End-to-End delay in the network is a priority, then CBS is a valuable tool. In our experiments, SV traffic was classified as ST and scheduled by HERMES. If jitter is of minor concern, SV traffic could instead be reclassified as AVB and scheduled online with CBS. Our experiments suggest that this approach will lower End-to-End delays in high traffic situations compared to non-TSN setups.

During our simulations, every scenario met its deadlines. However, it is possible that other use cases or scenarios might not always do so. For the scenarios using TSN combined with CBS, we can guarantee that deadlines will be met for SV and GOOSE messages. This assurance is based on the schedules created by the HERMES scheduler and confirmed through the AVB analysis, which shows that all AVB messages meet their deadlines. It is crucial to note that one of the main advantages of TSN is considered to be its ability to guarantee the arrival of all time-critical traffic and its reliability in managing time-sensitive network traffic [21], [22].

In regards to RQ1, our results show that all scenarios manage to deliver all messages within their respective deadlines. Our experiments indicate that TSN might reduce End-to-End delay for messages using the IEC 61850 process bus during high traffic loads by utilising AVB classification for high-priority traffic. To further support these conclusions, additional use cases and scenarios should be evaluated.

9.2 Jitter

Since we classify SV traffic as ST, we expect it to be more predictable in the TSN scenarios compared to the Ethernet scenarios. By scheduling traffic into designated timeslots, we can potentially eliminate jitter entirely. In our simulations, detailed in Section 8.2, we opted to investigate the instantaneous jitter. From analysing our results in Figures 16 to 18, and by revisiting specific data points through repeated simulation runs, we observed that in Ethernet scenarios without prioritised traffic, SV jitter primarily results from SV messages competing for bandwidth with logging and MMS traffic. Consequently, SV messages, despite their time-critical nature, can get queued behind MMS and log messages. Implementing priority queuing can mitigate this issue to some extent, as shown in Figure 19, where the maximum jitter in high-load Ethernet scenarios decreased from 121 µs to 65 µs. However, the mean jitter is only marginally reduced because other types of traffic still interfere in the outgoing queues. A message that is currently being sent cannot be pre-empted in our use case.

When utilising TSN, the mean jitter is significantly lower than Ethernet scenarios under high traffic load, as bandwidth is reserved in specific time slots in a repeating schedule. This ensures that most SV messages experience nearly identical end-to-end delays. However, it is important to note that the maximum jitter in TSN scenarios remains relatively high compared to Ethernet scenarios. Upon further analysis, we found that the scheduler HERMES did not design the schedule to be completely jitter-free but instead focused on meeting deadlines. However, this approach has still significantly reduced the mean jitter in high-load TSN scenarios while also decreasing the maximum jitter. Furthermore, it is theoretically possible to tighten scheduler constraints to create a jitter-free environment.

By examining Figure 18, it is evident that the jitter is typically very close to 0 µs, but it often spikes to either 12 µs or -12 µs. This pattern can be attributed to the interaction between the scheduling system and clock synchronisation. Since the SV messages are produced simultaneously by two different sources, the MUs, they arrive at the switch at the same time. Here, they are placed in the highest priority queue to await the TAS to open the gate. Given that both messages are of the same priority and the queue operates on a First In, First Out (FIFO) basis, the message that gets processed first depends on the synchronisation of the sender nodes. This could result in either MU A or MU B sending its message first. If this variability in jitter is undesirable, it can be mitigated by offsetting or adjusting the production interval of one source so messages are not generated at the same time, thus preventing jitter caused by synchronisation discrepancies.

Regarding RQ2, the findings of our studies show that the jitter can be significantly improved in an IEC 61850 process bus by employing TSN in favour of standard Ethernet. Nevertheless, further investigation is necessary using more detailed and realistic scenarios to identify the most effective TSN configurations for digital substations.

9.3 Clock Synchronisation

The 802.1AS standard guarantees that time-aware systems with up to six hops are synchronised within 1 µs [45]. In this thesis, our simulations across various scenarios confirm this synchronisation accuracy for our specific use case. However, since we have classified the time-triggered gPTP messages as ST, it has become evident that the scheduler can negatively impact synchronisation. In our experiments, HERMES schedules messages close to the end of the period in TSN scenarios. This scheduling approach is likely the cause of gPTP synchronisation suffering compared to non-TSN scenarios. The schedule produces *Sync* and *pDelay* messages that arrive at the destination at the end of the period rather than at the beginning. Further experiments are required to validate the identified reasons for the errors. Nevertheless, synchronisation in the TSN scenarios still remains well within the 1 µs requirement. Furthermore, gPTP messages benefit from being classified in a protected class. Utilising various schedulers and higher traffic loads can aid in investigating whether TSN can positively affect clock synchronisation in a digital substation environment.

9.4 Final Reflections and Future Work

After performing simulations and analysing the results, our main insight is that employing TSN across various traffic classes can allow for different quality of service and behaviours, which is not achievable with strict priority queuing. However, employing TSN on a process bus is complex and demands a deep understanding of traffic characteristics and desired traffic behaviours. The level of jitter depends on our scheduling choices, a scheduler aimed at reducing jitter while creating a feasible schedule might increase the End-to-End delay due to these constraints. For traffic classified as ST, which we schedule offline, our focus is generally not on minimising End-to-End delay but rather on ensuring that deadlines are met. This was particularly relevant for SV traffic in our use case. Nonetheless, if reducing End-to-End delay is a priority and jitter is less of a concern, reclassifying SV as AVB might be beneficial to achieve lower latencies.

This thesis acknowledges the limitation that simulations do not perfectly mirror real-world conditions. Additionally, a large number of simulation runs are needed to demonstrate confidence in the resulting trends. However, due to hardware limitations, we were only able to run 50 simulations per scenario. Despite this, the lack of outliers in the results indicates that the number of experiments conducted is sufficient to draw meaningful conclusions.

Moreover, our limited understanding of actual traffic flow within a process bus might have led to somewhat unrealistic scenarios concerning traffic volume and routing. Future scenarios should incorporate more realistic traffic loads and patterns, as well as scenarios that result in higher link congestion. Our studies mainly involved overloading the network with TCP traffic during the high-load scenarios. TCP utilises sliding windows to adjust transmission rates automatically, thus avoiding packet queue overloads. Introducing non-TCP traffic types could overload the switch queues more and facilitate packet drops, and the effects of utilising TSN may be shown more clearly.

10 Conclusions

In this thesis, we have investigated the effects on latency, jitter and clock synchronisation in an IEC 61850 Process Bus when allocating messages into different TSN classes. Several simulation scenarios have been implemented and evaluated. The scenarios have consisted of different networks using Ethernet without priority queuing, Ethernet with priority queuing, and TSN utilising TAS and CBS. The scenarios have also been simulated during low and high traffic loads. Considering the critical role of digital substations in power grids, the study aimed to explore the potential benefits of TSN, and the possibility to enhance the reliability and efficiency of digital substation networks.

We have concluded that TSN can positively impact the performance of an IEC 61850 Process Bus. Our findings demonstrate that End-to-End delay can be improved by classifying traffic as AVB and utilising CBS. Furthermore, jitter has been significantly improved for SV traffic. With TSN it is also possible to give hard real-time deadline guarantees by classifying messages as ST and performing a response time analysis for AVB traffic. However, configuring a TSN network in a digital substation setting is not trivial and requires an understanding of the traffic patterns inside the digital substation and process bus. To achieve the desired results, careful consideration must be given to how traffic is classified into TSN classes and the selection of an appropriate scheduling mechanism. For AVB traffic specifically, a traffic analysis is required to reach an appropriate idleSlope value for CBS.

The significance of our results lies in their implications for the design and management of future digital substations. By utilising TSN and scheduling traffic effectively, alongside analysing AVB traffic, digital substations can manage increased traffic loads without compromising the delivery of time-critical data.

References

[1] Hector Leon, Carlos Montez, Marcelo Stemmer, and Francisco Vasques. Simulation models for IEC 61850 communication in electrical substations using GOOSE and SMV time-critical messages. In *2016 IEEE World Conference on Factory Communication Systems (WFCS)*, pages 1–8, Aveiro, Portugal, May 2016. IEEE. ISBN 978-1-5090-2339-4. doi: 10.1109/WFCS.2016.7496500. URL `http://ieeexplore.ieee.org/document/7496500/`.

[2] Jonathan Falk, David Hellmanns, Ben Carabelli, Naresh Nayak, Frank Durr, Stephan Kehrer, and Kurt Rothermel. NeSTiNg: Simulating IEEE Time-sensitive Networking (TSN) in OMNeT++. In *2019 International Conference on Networked Systems (NetSys)*, pages 1–8, Munich, Germany, March 2019. IEEE. ISBN 978-1-72810-568-0. doi: 10.1109/NetSys.2019.8854500. URL `https://ieeexplore.ieee.org/document/8854500/`.

[3] OMNeT++ team. Time-sensitive networking (tsn) introduced in inet, 2022. URL `https://inet.omnetpp.org/2022-06-15-tsn-released.html`. Accessed: 2024-01-25.

[4] Théo Docquier, Ye-Qiong Song, and Vincent Chevrier. On the relevance of TSN for Substation Communication Networks. In *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–8, September 2022. doi: 10.1109/ETFA52439.2022.9921471. URL `https://ieeexplore.ieee.org/document/9921471`.

[5] Théo Docquier, Ye-Qiong Song, Vincent Chevrier, Ludovic Pontnau, and Abdelaziz Ahmed-Nacer. IEC 61850 over TSN: traffic mapping and delay analysis of GOOSE traffic. In *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, volume 1, pages 246–253, September 2020. doi: 10.1109/ETFA46521.2020.9212159. URL `https://ieeexplore.ieee.org/document/9212159`. ISSN: 1946-0759.

[6] Jay F. Nunamaker, Minder Chen, and Titus D. M. Purdin. Systems Development in Information Systems Research. *Journal of Management Information Systems*, 7(3):89–106, 1990. ISSN 0742-1222. URL `https://www.jstor.org/stable/40397957`. Publisher: Taylor & Francis, Ltd.

[7] Archana T. What is Electrical Substation? Definition, Single Line Diagram & Site Selection, December 2016. URL `https://circuitglobe.com/electrical-substation.html`. Accessed: 2024-01-17.

[8] What is an electricity substation? | National Grid Group, January 2024. URL `https://www.nationalgrid.com/stories/energy-explained/what-is-a-substation`. Accessed: 2024-01-17.

[9] Rahul Gore, Hariram Satheesh, Mahesh Varier, and Simi Valsan. Analysis of an IEC 61850 based Electric Substation Communication Architecture. In *2016 7th International Conference on Intelligent Systems, Modelling and Simulation (ISMS)*, pages 388–393, Bangkok, Thailand, January 2016. IEEE. ISBN 978-1-5090-0665-6. doi: 10.1109/ISMS.2016.85. URL `http://ieeexplore.ieee.org/document/7877244/`.

[10] ABB. How digital substations tame the complexities of modern power grids. `https://library.abb.com/d/4CAE000922`, 2020. Accessed: 2024-01-19.

[11] Anupam Chattopadhyay, Abhisek Ukil, Dirmanto Jap, and Shivam Bhasin. Toward threat of implementation attacks on substation security: Case study on fault detection and isolation. *IEEE Transactions on Industrial Informatics*, 14(6):2442–2451, 2018. doi: 10.1109/TII.2017.2770096.

[12] Ching-Lai Hor and P.A. Crossley. Knowledge Extraction from Intelligent Electronic Devices. *Transactions on Rough Sets*, 3400:82–111, January 2005. ISSN 978-3-540-25998-5. doi: 10.1007/11427834_4.

[13] Rahul N. Gore, Elena Lisova, Johan Åkerberg, and Mats Björkman. In Sync with Today's Industrial System Clocks. In *2020 International Conference on COMmunication Systems & NETworkS (COMSNETS)*, pages 785–790, Bengaluru, India, January 2020. IEEE. ISBN 978-1-72813-187-0. doi: 10.1109/COMSNETS48256.2020.9027323. URL `https://ieeexplore.ieee.org/document/9027323/`.

[14] Dung Le Tien, Quang Hoang Ngoc Hoai, and Minh Kieu Van. Experience With A Digital Substation Pilot Project of Vietnam Electricity. In *2022 11th International Conference on Control, Automation and Information Sciences (ICCAIS)*, pages 1–6, November 2022. doi: 10.1109/ICCAIS56082.2022.9990264. URL `https://ieeexplore.ieee.org/document/9990264`. ISSN: 2475-7896.

[15] Mahmood Qureshi, Ali Raza, Dileep Kumar, Sang-Sig Kim, Un-Sig Song, Min-Woo Park, Hyuk-Soo Jang, Hyo-Sik Yang, and Byung-Seok Park. A survey of communication network paradigms for substation automation. In *2008 IEEE International Symposium on Power Line Communications and Its Applications*, pages 310–315, Jeju City, Jeju Island, South Korea, April 2008. IEEE. ISBN 978-1-4244-1975-3 978-1-4244-1976-0. doi: 10.1109/ISPLC.2008.4510445. URL `http://ieeexplore.ieee.org/document/4510445/`.

[16] R.E. Mackiewicz. Overview of IEC 61850 and Benefits. In *2006 IEEE PES Power Systems Conference and Exposition*, pages 623–630, Atlanta, Georgia, USA, 2006. IEEE. ISBN 978-1-4244-0177-2. doi: 10.1109/PSCE.2006.296392. URL `http://ieeexplore.ieee.org/document/4075831/`.

[17] IEC TR 61850-90-13:2021 - Communication networks and systems for power utility automation - Part 90-13: Deterministic networking technologies. Technical Report IEC TR 61850-90-13:2021, International Electrotechnical Commission, Geneva, Switzerland, 2021.

[18] Paolo Ferrari, Alessandra Flammini, Stefano Rinaldi, and Gunnar Prytz. Time synchronization concerns in substation automation system. In *2011 IEEE International Workshop on Applied Measurements for Power Systems (AMPS)*, pages 112–117, Aachen, Germany, September 2011. IEEE. ISBN 978-1-61284-947-8 978-1-61284-946-1 978-1-61284-945-4. doi: 10.1109/AMPS.2011.6090431. URL `http://ieeexplore.ieee.org/document/6090431/`.

[19] Ieee standard for a precision clock synchronization protocol for networked measurement and control systems. *IEEE Std 1588-2019 (Revision of IEEE Std 1588-2008)*, pages 1–499, 2020. doi: 10.1109/IEEESTD.2020.9120376.

[20] Andre Vallat and Dominik Schneuwly. Clock synchronization in telecommunications via ptp (ieee 1588). In *2007 IEEE International Frequency Control Symposium Joint with the 21st European Frequency and Time Forum*, pages 334–341, 2007. doi: 10.1109/FREQ.2007.4319093.

[21] Lucia Lo Bello and Wilfried Steiner. A Perspective on IEEE Time-Sensitive Networking for Industrial Communication and Automation Systems. *Proceedings of the IEEE*, 107(6):1094–1120, June 2019. ISSN 0018-9219, 1558-2256. doi: 10.1109/JPROC.2019.2905334. URL `https://ieeexplore.ieee.org/document/8695835/`.

[22] Youhwan Seol, Doyeon Hyeon, Junhong Min, Moonbeom Kim, and Jeongyeup Paek. Timely Survey of Time-Sensitive Networking: Past and Future Directions. *IEEE Access*, 9:142506–142527, 2021. ISSN 2169-3536. doi: 10.1109/ACCESS.2021.3120769. URL `https://ieeexplore.ieee.org/document/9576720/`.

[23] F. Zezulka, P. Marcon, Z. Bradac, J. Arm, and T. Benesl. Time-Sensitive Networking as the Communication Future of Industry 4.0. *IFAC-PapersOnLine*, 52(27):133–138, January 2019. ISSN 2405-8963. doi: 10.1016/j.ifacol.2019.12.745. URL `https://www.sciencedirect.com/science/article/pii/S2405896319326941`.

[24] IEEE Standard for Local and Metropolitan Area Networks–Bridges and Bridged Networks. *IEEE Std 802.1Q-2022 (Revision of IEEE Std 802.1Q-2018)*, pages 1–2163, December 2022. doi: 10.1109/IEEESTD.2022.10004498. URL `https://ieeexplore.ieee.org/document/10004498`. Conference Name: IEEE Std 802.1Q-2022 (Revision of IEEE Std 802.1Q-2018).

[25] IEEE Standard for Local and metropolitan area networks - Station and Media Access Control Connectivity Discovery. *IEEE Std 802.1AB-2016 (Revision of IEEE Std 802.1AB-2009)*, pages 1–146, March 2016. doi: 10.1109/IEEESTD.2016.7433915. URL `https://ieeexplore.ieee.org/document/7433915`. Conference Name: IEEE Std 802.1AB-2016 (Revision of IEEE Std 802.1AB-2009).

[26] IEEE Standard for Local and Metropolitan Area Networks–Timing and Synchronization for Time-Sensitive Applications. *IEEE Std 802.1AS-2020 (Revision of IEEE Std 802.1AS-2011)*, pages 1–421, June 2020. doi: 10.1109/IEEESTD.2020.9121845. URL `https://ieeexplore.ieee.org/document/9121845`. Conference Name: IEEE Std 802.1AS-2020 (Revision of IEEE Std 802.1AS-2011).

[27] IEEE Standard for Local and Metropolitan Area Networks–Link Aggregation. *IEEE Std 802.1AX-2020 (Revision of IEEE Std 802.1AX-2014)*, pages 1–333, May 2020. doi: 10.1109/IEEESTD.2020.9105034. URL `https://ieeexplore.ieee.org/document/9105034`. Conference Name: IEEE Std 802.1AX-2020 (Revision of IEEE Std 802.1AX-2014).

[28] IEEE Standard for Local and metropolitan area networks–Frame Replication and Elimination for Reliability. *IEEE Std 802.1CB-2017*, pages 1–102, October 2017. doi: 10.1109/IEEESTD.2017.8091139. URL `https://ieeexplore.ieee.org/document/8091139`. Conference Name: IEEE Std 802.1CB-2017.

[29] IEEE Standard for Local and Metropolitan Area Networks–Link-local Registration Protocol. *IEEE Std 802.1CS-2020*, pages 1–151, April 2021. doi: 10.1109/IEEESTD.2021.9416320. URL `https://ieeexplore.ieee.org/document/9416320`. Conference Name: IEEE Std 802.1CS-2020.

[30] Ahmed Nasrallah, Akhilesh S. Thyagaturu, Ziyad Alharbi, Cuixiang Wang, Xing Shao, Martin Reisslein, and Hesham Elbakoury. Performance Comparison of IEEE 802.1 TSN Time Aware Shaper (TAS) and Asynchronous Traffic Shaper (ATS). *IEEE Access*, 7:44165–44181, 2019. ISSN 2169-3536. doi: 10.1109/ACCESS.2019.2908613. URL `https://ieeexplore.ieee.org/document/8681083`. Conference Name: IEEE Access.

[31] Jingyue Cao, Mohammad Ashjaei, Pieter J. L. Cuijpers, Reinder J. Bril, and Johan J. Lukkien. An independent yet efficient analysis of bandwidth reservation for credit-based shaping. In *2018 14th IEEE International Workshop on Factory Communication Systems (WFCS)*, pages 1–10, June 2018. doi: 10.1109/WFCS.2018.8402345. URL `https://ieeexplore.ieee.org/document/8402345`.

[32] Bahar Houtan. Configuring and Analysing TSN Networks Considering Low-priority Traffic. 2021. URL `https://urn.kb.se/resolve?urn=urn:nbn:se:mdh:diva-56349`. Publisher: Mälardalen university.

[33] Rubi Debnath, Philipp Hortig, Luxi Zhao, and Sebastian Steinhorst. Advanced Modeling and Analysis of Individual and Combined TSN Shapers in OMNeT++. In *2023 IEEE 29th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, pages 176–185, August 2023. doi: 10.1109/RTCSA58653.2023.00029. URL `https://ieeexplore.ieee.org/abstract/document/10296445`. ISSN: 2325-1301.

[34] Théo Docquier, Ye-Qiong Song, Vincent Chevrier, Ludovic Pontnau, and Abdelaziz Ahmed-Nacer. Generating substation network simulations from substation configuration description files. In *13th Junior Researcher Workshop on Real-Time Computing (JRWRTC)*, 2019.

[35] Mitalkumar G. Kanabar and Tarlochan S. Sidhu. Performance of IEC 61850-9-2 Process Bus and Corrective Measure for Digital Relaying. *IEEE Trans. Power Delivery*, 26(2):725–735, April 2011. ISSN 0885-8977, 1937-4208. doi: 10.1109/TPWRD.2009.2038702. URL `http://ieeexplore.ieee.org/document/5409526/`.

[36] Tarlochan S. Sidhu and Yujie Yin. Ied modelling for iec 61850 based substation automation system performance simulation. In *Power Engineering Society General Meeting*, pages 7–pp. IEEE, 2006.

[37] Tarlochan S. Sidhu and Yujie Yin. Modelling and simulation for performance evaluation of iec 61850-based substation communication systems. *IEEE Transactions on Power Delivery*, 22(3):1482–1489, 2007.

[38] Héctor León, Carlos Montez, Odilson Valle, and Francisco Vasques. Real-Time Analysis of Time-Critical Messages in IEC 61850 Electrical Substation Communication Systems. *Energies*, 12:2272, June 2019. doi: 10.3390/en12122272.

[39] Mohammad Golshani, Gareth A. Taylor, and Ioana Pisica. Simulation of power system substation communications architecture based on IEC 61850 standard. In *2014 49th International Universities Power Engineering Conference (UPEC)*, pages 1–6, Cluj-Napoca, Romania, September 2014. IEEE. ISBN 978-1-4799-6557-1 978-1-4799-6556-4. doi: 10.1109/UPEC.2014.6934745. URL `http://ieeexplore.ieee.org/document/6934745/`.

[40] Jorge Sanchez-Garrido, Antonio Jurado, Luis Medina, Rafael Rodriguez, Eduardo Ros, and Javier Diaz. Digital Electrical Substation Communications Based on Deterministic Time-Sensitive Networking Over Ethernet. *IEEE Access*, 8:93621–93634, 2020. ISSN 2169-3536. doi: 10.1109/ACCESS.2020.2995189. URL `https://ieeexplore.ieee.org/document/9095217/`.

[41] Daniel Bujosa, Mohammad Ashjaei, Alessandro V. Papadopoulos, Thomas Nolte, and Julian Proenza. HERMES: Heuristic Multi-queue Scheduler for TSN Time-Triggered Traffic with Zero Reception Jitter Capabilities. In *Proceedings of the 30th International Conference on Real-Time Networks and Systems*, RTNS ’22, pages 70–80, New York, NY, USA, June 2022. Association for Computing Machinery. ISBN 978-1-4503-9650-9. doi: 10.1145/3534879.3534906. URL `https://dl.acm.org/doi/10.1145/3534879.3534906`.

[42] Mohammad Ashjaei, Gaetano Patti, Moris Behnam, Thomas Nolte, Giuliana Alderisi, and Lucia Lo Bello. Schedulability analysis of Ethernet Audio Video Bridging networks with scheduled traffic support. *Real-Time Systems*, 53(4):526–577, July 2017. ISSN 1573-1383. doi: 10.1007/s11241-017-9268-5. URL `https://doi.org/10.1007/s11241-017-9268-5`.

[43] Lucas Santos Dalenogare, Guilherme Brittes Benitez, Néstor Fabián Ayala, and Alejandro Germán Frank. The expected contribution of industry 4.0 technologies for industrial performance. *International Journal of production economics*, 204:383–394, 2018.

[44] IEC 61850-5:2013-01+AMD1:2022-03 - Communication networks and systems for power utility automation Part 5: Communication requirements for functions and device models, 2022.

[45] Marina Gutierrez, Wilfried Steiner, Radu Dobrin, and Sasikumar Punnekkat. Synchronization Quality of IEEE 802.1AS in Large-Scale Industrial Automation Networks. In *2017 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 273–282, Pittsburg, PA, USA, April 2017. IEEE. ISBN 978-1-5090-5269-1. doi: 10.1109/RTAS.2017.10. URL `http://ieeexplore.ieee.org/document/7939046/`.