Time Synchronization for Large Volume Metrology in Industrial Networks

Lappeenranta–Lahti University of Technology LUT

Master's Programme in electrical engineering, master's thesis

2023

Hesam Shaygan

Examiner(s): Associate Professor Pedro Nardelli
Daniel Gutierrez Rojas D.Sc. (Tech.)

ABSTRACT

Lappeenranta–Lahti University of Technology LUT

LUT School of Energy Systems

Electrical Engineering

Hesam Shaygan

Time Synchronization for Large Volume Metrology in Industrial Networks

Master's thesis

2023

51 pages, 11 figures, 1 table

Examiner(s): Professor Pedro Nardelli and Daniel Gutierrez Rojas, D.Sc. (Tech.)

Keywords: Time Synchronization, Industrial Networks, LVM, PTP (Precision Time Protocol), gPTP (Generalized Precision Time Protocol), White Rabbit, EtherCAT, PowerLink, ProfiNET IRT, TSN (Time-Sensitive Networking), OPC UA TSN (OPC Unified Architecture Time-Sensitive Networking), URLLC (Ultra-Reliable Low Latency Communications) in 5G, Intel i210 Card, Linux System, Network Performance, Metrology, NTP (Network Time Protocol).

This thesis investigates time synchronization and real-time communication in large networks, particularly focusing on their application within industrial environments. The study utilizes the ptp4l program, an industrial protocol widely used for time synchronization purposes, to gain insights into the intricacies of time synchronization processes. The performances of two setups, namely the White Rabbit and Perle IDS-305, are evaluated through oscilloscope and laser tests.

Findings from this research contribute to a better understanding of time synchronization in large networks and can aid in decision-making processes when choosing between different setups. However, the results are limited to the specific devices and tests used in this study. Future research could explore different time synchronization methods and devices and attempt to optimize time synchronization in large networks.

ACKNOWLEDGEMENTS

I'm immensely grateful for the guidance and support I've received throughout the course of this thesis. My supervisors at VTT, Mr. Jeremias Seppa and Mr. Antti Lassila, have played an instrumental role in shaping this work with their invaluable insights and unending patience. Their commitment to excellence has driven me to delve deeper into the complexities of the subject and I sincerely appreciate their belief in my abilities. Equally, I'm thankful for the collaborative spirit and assistance provided by my colleagues at VTT, Mr. Kalle Hanhijarvi and Mr. Tapoi Lehtonen. Their expertise has significantly enhanced the breadth of this work.

At Lappeenranta-Lahti University of Technology (LUT), I've been fortunate to have Prof. Pedro Nardelli as my supervisor. His profound knowledge and clear direction have helped me maintain a focused approach throughout. Additionally, the constructive criticism and insightful feedback from Daniel Gutierrez Rojas, D.Sc. (tech.), the examiner of this thesis, have greatly contributed towards refining the final output. Above all, I want to acknowledge the encouraging and conducive environment provided by both LUT and VTT Technical Research Centre of Finland Ltd, which fostered the successful completion of my master's thesis. Lastly, but certainly not least, I express my deepest gratitude towards my family and friends for their unyielding support and unwavering belief in me.

Table of contents

Abstract

Acknowledgements

Table of Contents

Figures

1 Introduction

1.1 Background and Context

Time synchronization and real-time communication in industrial networks are critical for ensuring the smooth operation of numerous automated and semi-automated processes. In industrial applications such as manufacturing and production, precise timing control is crucial for coordinating various elements within the system, including sensors, actuators, and other embedded devices (Gore et al., 2020).

The concept of a Large Volume Metrology (LVM) network becomes important in the context of multi-party manufacturing networks. Measurement of objects with diameters between a few and tens of meters is the focus of the discipline of LVM (Carmignato et al., 2013). Large-scale industries including automotive, aerospace, and energy have seen increased precision because of LVM improvements (Carmignato et al., 2013). The precision and dependability of LVM have substantially increased with the development of laser trackers and other cutting-edge measuring technologies (Hockenberry et al., 2015).

However, there are numerous difficulties in achieving accurate time synchronization and real-time communication in these networks. The accuracy of the synchronization might be impacted by variables such as network latency, clock drift, and jitter. Furthermore, many devices, each with a unique internal clock operating at a slightly different speed, might exacerbate synchronization issues (Gore et al., 2020).

Real-time communication deployment in industrial networks is a difficult task as well. For instance, due to their innate vulnerability to interference, signal fading, and multi-path propagation, wireless mesh networks present difficulties (Han et al., 2011). A mix of efficient network design and strong protocols is required to ensure dependable and real-time communication in such contexts (Han et al., 2011).

For a variety of reasons, it is imperative to address these issues. Coordination of operations across a distributed network is made possible by precise time synchronization, ensuring effectiveness, and preventing potential collisions or conflicts (Gore et al., 2020). For monitoring and control applications, on the other hand, real-time communication is essential since delays can result in severe performance degradation or even catastrophic failures (Han et al., 2011).

1.2 Purpose and Research Questions

This thesis aims to evaluate approaches for time synchronization and real-time communication in LVM networks within multi-party factory settings.

To reach this objective, the following research questions need to be answered:

1. What are the existing and emerging approaches for time synchronization in industrial environments, such as Precision Time Protocol (PTP), Generalized Precision Time Protocol (gPTP), and WhiteRabbit?
2. How can real-time capabilities be effectively incorporated into Large Volume Metrology (LVM) networks within multi-party factory settings, considering emerging standards like Time-Sensitive Networking (TSN), OPC UA TSN, and the Ultra-Reliable Low Latency Communication (URLLC) capabilities of 5G?
3. What is the performance and reliability of a PTP system utilizing an Intel i210 card on Linux, assessed through oscilloscope and laser-based testing methodologies?

Insights from this study should help improve real-time communication and time synchronization in industrial networks.

1.3 Thesis Structure

This thesis is organized into seven comprehensive chapters, each serving a distinctive purpose:

Chapter 1: Introduction

In this chapter, the study topic is introduced, the problem is stated, and the research questions and objectives are formulated. As well as introducing the reader to the thesis's form, it defines the research's scope and importance.

Chapter 2: Literature Review

This chapter surveys the literature on field bus protocols based on Ethernet, real-time communication in industrial networks, and time synchronization methods. It establishes a theoretical foundation for the investigation and points out knowledge gaps that the thesis seeks to address.

Chapter 3: Methodology

The research methodology, data-gathering procedures, and analytic strategies are all covered in this chapter. It also describes the metrics for evaluating various time synchronization techniques and real-time communication tactics.

Chapter 4: PTP System Implementation and Testing

The PTP system's implementation and the techniques required to evaluate it using an oscilloscope and a laser are covered in length in this chapter.

Chapter 5: Results and Discussion

The findings of the oscilloscope and laser tests are presented in this chapter. It also contains a thorough study and comparison of the two sets of results, as well as a discussion of the conclusions' implications for LVM networks.

Chapter 6: Conclusion and Future Work

This chapter highlights the research contributions, summarizes the key research findings, and notes the study's shortcomings. Additionally, it offers helpful advice and proposes areas for further study.

2 Literature Review

This chapter reviews the research on Ethernet-based field bus protocols, real-time industrial network communication, and time synchronization techniques. It identifies knowledge gaps that the thesis aims to fill and provides a theoretical framework for the inquiry.

2.1 Metrology in time synchronization

Metrology, the science of measurement, is fundamental to accurate time synchronization across diverse systems and industries (Matsakis, Levine, & Lombardi, 2018). The precision of these measurements is crucial to the seamless operation of these systems, and any discrepancies could lead to substantial complications (Carmignato, Savio, & Savio, 2013). The Network Time Protocol (NTP) applies metrology techniques to provide precise timekeeping to networked devices (Mills, 1991). It calculates the round-trip delay time between a client and server to determine and correct any time offset. The accuracy of these measurements is vital for maintaining correct time synchronization across networks (Mills, 1991The Precision Time Protocol (PTP) is frequently used in industrial settings, especially in automation where high accuracy time synchronization is required (Eidson, 2006). In this application, metrology refers to the calibration of measurement instruments to a recognized standard in order to ensure precise synchronization. This accuracy is crucial for coordinating real-time operations, such as assembly lines and other automated processes (Carmignato et al., 2013; Eidson, 2006).

The IEEE 1588 standard, which describes PTP, exemplifies the application of metrology in enhancing time synchronization in industrial settings (Eidson, 2006). It utilizes metrology techniques to account for network delays and variabilities, aiming to provide precise time coordination of real-time events across networks. Achieving synchronization accuracy within the nanoseconds range is one of the main goals of this standard (Eidson, 2006).

In telecommunications, timing synchronization is critical, such as in Long-Term Evolution (LTE) and 5G networks (Bregni, 2002). Metrology techniques are deployed to ensure the required accuracy within tens of nanoseconds (Bregni, 2002; Agustoni et al., 2022).

In summary, metrology plays a significant role in maintaining accurate time synchronization in various domains, from computer networks to industrial settings and telecommunication systems. By providing the scientific basis for precise measurements, metrology ensures the reliability, efficiency, and coordination of systems dependent on precise timekeeping (Carmignato et al., 2013; Hockenberry, Song, & Stone, 2015).

2.2 Time Synchronization in Industrial Networks

In industrial networks, such as large-volume metrology (LVM) networks, where precise and synchronized timing across all network nodes is of utmost importance, time synchronization is vital (UKEssays, 2018). Real-time control systems, where it permits exact coordination between many components, are one of the most significant uses of time synchronization in industrial networks (Thakur & Satish, 2022). This is particularly important for LVM networks since they depend on accurate measurements and data. Without precise time synchronization, data processing inconsistencies could happen, producing false findings and ineffective performance. The idea of time synchronization is lining up the clocks of several gadgets connected to a network so that they run simultaneously (Sampath & Tripti, 2010). It guarantees that all actions and procedures are carried out promptly and efficiently. Data inconsistencies, communication issues, and general system dysfunction could result from a lack of synchronization.

With the introduction of Industrial Internet of Things (IIoT) applications, the significance of clock synchronization in industrial networks, including LVM, is further increased. Precision Time Protocol (PTP), also known as IEEE 1588, a clock synchronization standard, has become crucial in this situation (IEEE, 2023). To achieve high levels of precision and dependability in IIoT systems, it provides a mechanism for precise clock synchronization in packet-based networks. But there are still problems to be solved. Future industrial networks' efficiency may be increased by clock synchronization, although (Gore et al.2020) point out that there are other considerations, such packet delay variation and clock drift. Robust synchronization methods and procedures are needed to ensure the industrial networks' seamless operation in the face of these issues.

2.2.1 Network Time Protocol (NTP)

The Network Time Protocol (NTP) is a networking standard for synchronizing computer clocks over packet-switched data networks. Since its first development in 1985, the protocol has been continuously used across the internet (Mills, 1991).

According to Mills (1991), NTP's robustness, scalability, and fault-tolerance are among its key characteristics. A stratum-organized, hierarchical, semi-layered structure of time sources is used by NTP. The stratum 0, which includes atomic clocks and other extremely precise timekeeping equipment, is at the top. The main time servers are stratum 1 servers, which are directly connected to stratum 0 devices. Higher strata servers are used for synchronization by lower strata servers (Mills, 1994). For the best clock selection, NTP employs a statistical approach known as the Marzullo's algorithm. Additionally, it uses a complex error estimation technique that can adjust to a variety of circumstances, such as network jitter or latency (Rangaswamy & Murthy, 2021).

The importance of NTP stems from the necessity of accurate and reliable timekeeping in a range of applications. A trustworthy synchronized time system is necessary for a variety of tasks, including distributed system coordination and maintaining precise timestamps for forensic or security investigations (Rangaswamy & Murthy, 2021). Furthermore, precise time synchronization is essential to ensuring uninterrupted service delivery in sectors with real-time applications, such as telecommunications and broadcasting (Mills, 1991).

The main advantages of NTP are its scalability, stability, and tolerance to single-point failures, which are made possible by its hierarchical time distribution design. Additionally, it provides reasonably precise synchronization over open networks and is open source, making it widely available and adaptable to a variety of requirements (Rangaswamy & Murthy, 2021). NTP, however, has some shortcomings. It is vulnerable to network issues like jitter and delay, which could reduce the precision of time synchronization. Additionally, it might not provide the incredibly accurate synchronization required by some industrial applications or scientific research (Mills, 1994).

2.2.2 Precision Time Protocol (PTP - IEEE 1588)

Precision Time Protocol (PTP), known as IEEE 1588, is a protocol employed for synchronising clocks across a computer network. It is appropriate for measurement and control systems on a local area network because it achieves clock precision in the sub-microsecond range (Idrees, Ahmad, & Khan, 2020). According to Yuan Kexin, Guo Xuebing, and Tian Jianying (2002), the PTP works on the premise of a "master" clock dispersing time information and "slave" clocks receiving it. PTP works by having the master and slave clocks communicate via a series of messages called sync, follow-up, delay request, and delay response. These communications give the slave clock the information it needs to calculate the one-way delay and synchronize its local clock with the master's (Idrees et al., 2020). In industrial networks, where exact time synchronization is required for several applications, such as data acquisition systems, automated industrial processes, and power utility networks, PTP is very important (Idrees et al., 2020). For time-sensitive applications, this protocol offers greater accuracy and precision compared to the earlier Network Time Protocol (NTP).

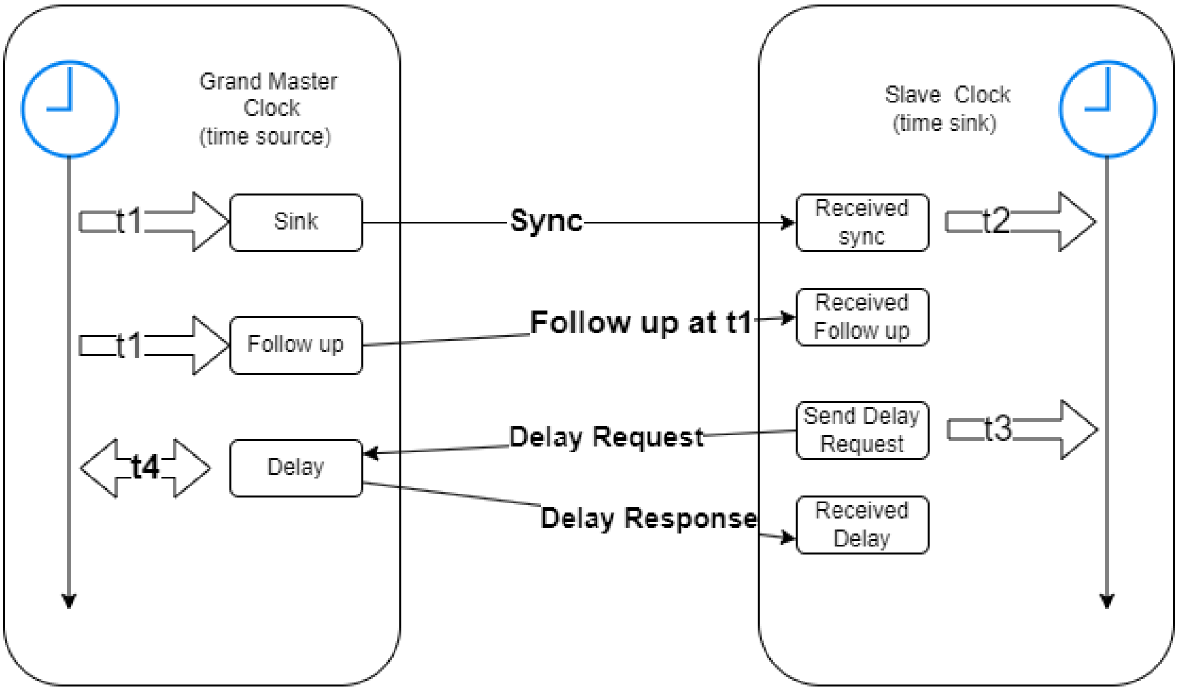

Figure 1: PTP message exchange algorithm. Adapted from (Idrees, Ahmad, & Khan, 2020)

PTP has many advantages in industrial applications due to its great precision. Data losses and system outages are decreased because of increased efficiency and dependability. Additionally, it permits synchronization across the entire network, which is crucial in expansive industrial networks (Wiesner & Kovácsházy, 2021).

PTP, however, also has several shortcomings. The accuracy of clock synchronization may be impacted by network delay variations even though PTP offers excellent accuracy (Rezabek, Helm, Leonhardt & Carle, n.d.). PTP's absence of built-in security safeguards makes it susceptible to hypothetical assaults that could prevent synchronization, which raises questions about its security (Rezabek et al., n.d.). PTP is used in several different sectors. By providing accurate time stamps for data logging and fault analysis, it is utilized, for example, in the power utility sector to guarantee the proper operation of devices and systems (Freire, Novaes, Almeida, Medeiros, Berg & Klautau, 2021). Additionally, PTP is used in the telecommunications sector to synchronize the clocks of various network components to ensure service quality (Idrees et al., 2020).

The Precision Time Protocol (PTP) necessitates the precise timestamping of packets, for which two main methods exist: hardware-based timestamping, also known as Ethernet header-based, and software-based timestamping, which can occur at the NIC (Network Interface Card), OS (Operating System) kernel, or user program level.

Hardware-based timestamping capitalises on a specialised clock within the hardware, typically the Ethernet hardware, providing a high degree of accuracy. The timestamps are recorded at the instant of packet transmission or reception, thereby minimising any delays associated with software processing (Idrees, Ahmad & Khan, 2020; Yuan Kexin, Guo Xuebing & Tian Jianying, 2021).

On the other hand, software-based timestamping presents a more flexible but potentially less precise approach. At the NIC level, the timestamping is done when the packet arrives at, or departs from, the NIC. When conducted at the OS kernel level, the timestamp is taken when the kernel processes the packet. Finally, user program level timestamping occurs within the application layer. Although adaptable, this method is most prone to latency and variability due to the potential influence of other system processes (Wiesner & Kovácsházy, 2021).

2.2.3 Generalized Precision Time Protocol (gPTP - IEEE 802.1 AS)

The generalized Precision Time Protocol (gPTP), as an extension of the Precision Time Protocol (PTP), provides significant enhancements in the distribution of time synchronization within networks, especially for time-sensitive applications. In the IEEE 802.1AS standard (IEEE 802.1AS-2011), gPTP is formally defined. The idea of a Grandmaster Clock, from which all other devices (referred to as slaves) in the network receive synchronization information, is maintained by gPTP, much like PTP. However, gPTP differs from PTP in several distinctive ways, with its key advantages being scalability, dependability, and suitability for time-sensitive networks. To ensure improved network resilience, gPTP utilizes a new Grandmaster Clock selection technique called the "best master clock," which enables dynamic adaptability and automatic reconfiguration (Fedullo et al., 2022).

The scalability that gPTP offers is another distinguishing quality, and it is essential for contemporary industrial and communication networks. When compared to PTP, the gPTP protocol reduces the amount of network traffic caused by timing messages, improving scalability in bigger networks (Lázaro et al., 2022).

Additionally, TSN (Time-Sensitive Networking), a collection of standards being developed by the IEEE 802.1 working group to deliver deterministic services across standard Ethernet, includes additional functions built expressly for it. For time-sensitive applications, these functionalities help to boost reliability, decrease latencies, and improve precision.

Many sectors that need precise time synchronization and network scalability have found extensive use for gPTP. The manufacturing sector, robotics, telecommunications, and multi-media broadcasting are all included in this, though not exclusively. For instance, gPTP's accurate and reliable synchronization is essential in Industrial Internet of Things (IIoT) contexts to effectively coordinate distributed measurement systems and smart devices (Fedullo et al., 2022). In multimedia broadcasting, gPTP guarantees precise synchronization for streaming services, which is necessary for applications that bridge audio and video.

Additionally, gPTP is crucial to Linux end equipment's implementation of the TSN protocol, guaranteeing accurate timing services and seamless connection with the current Ethernet infrastructure (Lázaro et al., 2022). gPTP has been applied to the establishment of dependable and scalable TSN in the field of scientific research, aiding in the development of adaptable research infrastructures like EnGINE (Rezabek et al., 2022).

2.2.4 White Rabbit

The White Rabbit (WR) protocol, developed under the auspices of CERN, is a technology that provides accurate time synchronization and data transfer over Ethernet-based networks. The WR system may be extended to hundreds of kilometers with only modest synchronization performance loss over long-distance fiber links, which are generally up to 10 km in length (Yang et al., 2022; Hennig & Hoover, 2021). It also delivers high precision and nanosecond-level synchronization accuracy. The WR system's synchronization accuracy at the sub-nanosecond level, network scalability, and predictable latency are among its key characteristics. It is also appropriate for applications like particle physics experiments and other sectors that need exact timing and synchronization because it can handle massive amounts of data flow (Hennig & Hoover, 2021; Neelam et al., 2021).

Applications of the WR protocol show off its capacity to provide high precision synchronization, ensuring that data acquisition systems in such environments capture data in a synchronized and accurate manner (Yang et al., 2022). One such application is the state acquisition system of high-energy physics experimental devices. Like how it has been essential for radiation detector readout electronics, WR time synchronization has increased the precision and dependability of the outcomes (Hennig & Hoover, 2021). In comparison to conventional synchronization techniques like Network Time Protocol (NTP), the WR protocol's main benefit is its better precision and accuracy (Rangaswamy & Murthy, 2021). Further highlighting its superiority are its robustness against network congestion and capacity to serve real-time applications (Rizzi et al., 2018). The high level of complexity required for the WR protocol's implementation, however, is a potential disadvantage because it makes it difficult to integrate new hardware and software into existing systems (Rizzi et al., 2018). In addition, WR implementation costs may be higher than those associated with conventional NTP solutions (Mills, 1991).

Despite these difficulties, the advantages of WR synchronization make it an appealing option for high-demand applications requiring high-precision timing. As more industries become aware of its potential benefits, its use is likely to increase.

2.2.5 Protocol Comparison

In distributed systems, time synchronization is essential for ensuring coherency, correctness, and functionality (Thakur & Satish, 2022; UKEssays, 2018). To accomplish these goals, several time synchronization protocols, each with unique strengths and disadvantages, have been proposed and put into use, including NTP, PTP, gPTP, and White Rabbit.

One of the earliest time synchronization protocols used in distributed systems is the Network Time Protocol (NTP) (Mills, 1991). According to Neagoe, Cristea, and Banica (2006), it provides robustness over large-scale networks and accounts for fluctuating network latency. Despite these benefits, NTP's software-based timestamps limit its precision, making it with improved precision made possible by hardware-based timestamping, Precision Time Protocol (PTP) was created as an improvement to NTP (Neagoe et al., 2006). Due to its sensitivity to network delay fluctuation, it enables for sub-microsecond accuracy but is better suited for local area networks (Zirngibl, Helm, & Stubbe, 2017; Wiesner & Kovácsházy, 2021).

A variant of PTP called Generalized Precision Time Protocol (gPTP) was created especially for industrial automation systems and offers additional improvements like better accuracy and synchronization (IEEE, 2023). unsuitable for applications that demand great accuracy (Rangaswamy & Murthy, 2021).

The standard PTP protocol gains synchronous Ethernet and accurate phase measurement capabilities with White Rabbit, a high-precision variant of PTP (Waterman, Helm, Zirngibl, & Stubbe, n.d.). White Rabbit is perfect for high-precision applications like high-energy physics experiments because it achieves synchronization accuracy in the sub-nanosecond range (Neelam et al., 2021; Hennig & Hoover, 2021; Yang et al., 2022).

Precision and performance of clock synchronization technology can be improved by using FPGA. As they may be adjusted for certain synchronization requirements and provide hardware-based timestamping, like PTP and White Rabbit, FPGA-based systems can offer flexibility and precision (Chen, Hao, & Xiaofeng, 2015).

By enabling deterministic communication over Ethernet networks, technologies like Time-Sensitive Networks (TSNs) make it easier to use these protocols in practical applications. The potential uses of synchronization protocols like PTP, gPTP, and White Rabbit have been shown to be expanded by the effective use of TSNs (Fedullo et al., 2022; Lázaro et al., 2022).

Rezabek, Helm, Leonhardt, and Carle (n.d.) found that security measures can also affect synchronization accuracy. Their research emphasizes the potential trade-off between security and precision in network synchronization by demonstrating how security measures may affect PTP synchronization accuracy.

In conclusion, NTP, PTP, gPTP, and White Rabbit each have special benefits. Large-scale networks can use NTP, LANs can benefit from PTP's high precision, industrial applications can use gPTP, and demanding situations can use White Rabbit's ultra-precise synchronization. The use of TSNs and FPGA technologies can improve these protocols' performance even more. The choice of an appropriate protocol is based on the needs of the application, considering elements like scalability, precision, and the balance between security and accuracy.

2.3 Real-Time Communication in Industrial Networks

In industrial networks, real-time communication is essential for coordinating and regulating multiple processes, assuring safety, and boosting productivity. The capacity to send and receive data in real-time is essential in a world where processes are becoming more automated and networked (Sundaresan & Bettati, 1997). LVM is the measuring of large-scale industrial settings, such as the shipbuilding, aircraft, or automobile industries. For the synchronization of numerous measuring equipment, the quality and consistency of data, and the real-time modification of processes based on these results, real-time communication is crucial in such situations (Doyle, 2004).

However, there are obstacles to real-time communication in industrial contexts. These consist of:

Network congestion and latency: Ensuring the swift and reliable transfer of data might be difficult in a congested industrial network. The physical distance that data must travel can add to latency problems, while network congestion can create delays (Sundaresan & Bettati, 1997).

Reliability and robustness: Industrial networks need to be robust and reliable, able to withstand harsh conditions and operate continuously without failure. It can be difficult to achieve this level of stability, particularly in wireless networks (Han et al., 2011).

Interoperability: Industrial networks frequently include a variety of equipment from different vendors. It can be difficult to ensure that these devices can properly communicate with one another in real-time (Ginthör et al., 2021).

2.3.1 Numerous Ethernet-based field bus protocols

Modern industrial networks, especially those that use real-time communication, must have Ethernet-based field bus protocols. They make it possible for systems and devices connected to the network to exchange data rapidly and effectively. EtherCAT®, PowerLink®, and ProfiNET IRT® have grown especially popular among the numerous Ethernet-based field bus protocols because of their benefits in speed, dependability, and interoperability with common Ethernet gear.

EtherCAT® (Ethernet for Control Automation Technology) is a highly regarded protocol known for its high-speed and low-latency performance. The EtherCAT Technology Group is responsible for overseeing the open standard EtherCAT®, which was created by Beckhoff Automation (Hibbard, 2016). With the ability to update up to 1,000 I/O points in just 30 microseconds, the protocol is famous for its exceptional performance. The time it takes to send, process, and receive data is predictable and repeatable thanks to EtherCAT®'s deterministic performance. In industrial automation settings, where precision and consistency are crucial, this quality is very valuable. EtherCAT® is not without its challenges, though. Due to its complex setup processes, implementing this protocol could be difficult.

PowerLink® presents another compelling option for real-time Ethernet-based field bus communication. The company behind PowerLink®, B&R Automation, created the protocol to provide deterministic communication over common Ethernet hardware (Hibbard, 2016). PowerLink® has a strong reputation and a sizable user base in the industrial and automation industries. The protocol is a great option for large networks with many nodes because it claims a network update rate of up to 400 microseconds. PowerLink® does, however, have a unique set of downsides. One is that the protocol needs particular master hardware to function, which could increase system costs.

ProfiNET IRT® (Industrial Real-Time) represents another evolution in high-speed, deterministic data transfer. ProfiNET IRT® was created by Siemens, a significant player in the industrial automation market, primarily for use in motion control operations and other applications requiring precise timing (Hibbard, 2016). Like its competitors, ProfiNET IRT® runs on common Ethernet gear and has cycle times as low as 250 microseconds. Despite having high performance, ProfiNET IRT®, like PowerLink®, requires particular master hardware. This criterion might affect overall costs and be viewed as a restriction.

All these protocols are Ethernet-based, however because of their diverse functionality and performance traits, they are better suited to various industrial applications. EtherCAT® is a solid option for systems that need determinism and quick I/O updates due to its high speed and efficiency. For bigger networks, PowerLink®, on the other hand, can provide a solid combination of performance and size scalability. Motion control is one area where ProfiNET IRT® excels that calls for accurate timing.

In the end, whatever protocol is preferred above the others depends on the requirements of the industrial network in issue. Which protocol is the greatest fit will depend on several factors, including the network's size, the necessary cycle time, hardware prices, and setup complexity (Hibbard, 2016). Despite having different characteristics, ProfiNET IRT®, PowerLink®, and EtherCAT® are all essential for providing effective, real-time communication in industrial contexts.

2.4 Emerging Standards

In the area of real-time industrial communication, we're seeing a growing need for ways to work better and faster. To help with this, new standards like OPC UA TSN are emerging (Pfrommer, Ebner, Ravikumar, & Karunakaran, 2018). This standard's 'publish/subscribe' mechanism is a crucial component. The Eugster, Felber, Guerraoui, and Kermarrec (2003) model makes complicated systems communicate more effectively.

The difficulties of industrial surroundings are also considered by these guidelines. For instance, they consider how the external environment may affect the system's timing accuracy (Schriegel & Jasperneite, 2007). These new standards, which seek to ensure that everything in a system is synchronized and functions properly together, must place a strong emphasis on timing. Research by the Industrial Internet Consortium (2018) has demonstrated the significance of these standards. Even with the known limitations of present technology, they can accommodate many types of data traffic on the same network, increasing the adaptability and efficiency of systems (Jasperneite, Schumacher, & Weber, 2007).

2.4.1 Time-Sensitive Networking (TSN - IEEE 802.1 Working Group)

Time-Sensitive Networking (TSN) has emerged as a significant development in the field of industrial automation and measurement systems. The revolutionary changes brought about by Industry 4.0, which changed the specifications for control and communication systems within the future production systems, have been the driving force behind this breakthrough (Fedullo et al 2022). The use of more sophisticated smart distributed measuring systems in industrial automation has prompted a wider adoption of the Industrial IoT (IIoT) paradigm, often known as the Internet of Things (IoT) paradigm. As they directly affect measurement accuracy, causation, dependability, and safety, communication networks and protocols are required to play important roles in this context (Fedullo et al 2022).

TSN came into focus as industrial communication systems, based on established technologies like Fieldbuses and Real-Time Ethernet (RTE) networks, faced challenges to meet the advanced applications' requirements. Among other things, these difficulties included tight time-criticality, high reliability, fault tolerance, and security. These issues are thought to be resolved by the IEEE 802.1 Time-Sensitive Networking (TSN) standardization project, which is acknowledged as the future industrial communications standard (Fedullo et al 2022). The TSN project aims to provide all the features needed to handle time-critical traffic in different scenarios. In recent years, there has been a steady increase in research interest in TSN development, particularly in the sectors of industrial automation and instrumentation and measurement (I&M). The limited implementation of TSN in the industrial situation, however, indicates that more research is required (Fedullo et al 2022). The primary standards under the TSN standardization activity are, among others, IEEE 802.1Qbv, which permits the transmission of time-critical traffic with strict latency requirements, IEEE 802.1Qcc, which offers a configuration model for the centralized control plane, IEEE 802.1CB, which supports the integration of fronthaul networks into the 5G network, and IEEE 802.1CM (Fedullo et al 2022).

Two profiles—IEEE 802.1Qcp and IEEE 802.1Qcj—have been made available for industrial automation applications in accordance with the TSN Industrial Automation Profile. The Qcp profile is designed for cyclic traffic and is rather straightforward. Joint Restricted (JrS), another name for the Qcj profile, is more complicated and supports a wider variety of applications. The creation of wireless capabilities for TSN and its incorporation into the Wi-Fi standard are some of the future research directions for this technology. In the context of industrial measurement, wireless communication presents a sizable possibility, and TSN has the potential to make a big impact in this field (Fedullo et al 2022).

2.4.2 OPC UA TSN

In the current industrial setting, OPC UA TSN (OPC Unified Architecture Time-Sensitive Networking) installation and development in manufacturing systems have important consequences. According to Gonzalez, Calderón, Figueiredo, and Sousa (2019), the technology has been hailed as a turning point for improving productivity, synchronization, and real-time data exchange.

The detailed study of the OPC Unified Architecture by Damm, Leitner, and Mahnke (2009) is provided. The authors explain the fundamental components of OPC UA while going over its uses, potentials, and difficulties in various settings. The book highlights OPC UA's versatility in terms of interoperability between various systems and platforms, reiterating the role it plays in enhancing system efficiency. The security characteristics that OPC UA TSN provides are a crucial part that Mahnke, Leitner, and Damm (2010) discuss. The authors contend that as industrial systems become more connected and complicated, security vulnerabilities become a major worry. They point out that OPC UA TSN is outfitted with strong security features, offering a layer of defense against potential online threats.

González et al. (2019) did a literature review and highlight how the OPC is used in advanced industrial settings. They acknowledge the OPC UA TSN's rising significance in integrating industrial devices and systems. The authors are grateful for how it facilitates smooth information flow across numerous platforms, enhancing operational effectiveness and data veracity. It is crucial to remember that obstacles still need to be overcome, notwithstanding OPC UA TSN's bright future. These include the difficulty of putting the technology into practice, the necessity for technical know-how, and the requirement for ongoing updates to maintain its effectiveness and security (Damm, Leitner, & Mahnke, 2009).

The operational effectiveness, data connectivity, and system security of OPC UA TSN have been demonstrated in manufacturing environments. It's a developing technology with a lot of potential, but to keep it secure and effective, it needs to be constantly monitored.

2.4.3 Ultra-Reliable Low-Latency Communications (URLLC) Capabilities of 5G

A key component of 5G and beyond networks is ultra-reliable and low-latency communication (URLLC), which aims to offer high reliability, low latency, and support for a variety of connected devices (Feng et al., 2021). With the introduction of URLLC, the landscape of wireless networks has undergone a substantial change, with several applications, difficulties, and opportunities emerging. Feng et al. (2021) investigated the uses, prospects, and difficulties of URLLC in their study. The authors explain the crucial function that URLLC performs in the IoT environment, where it is essential to manage a huge number of devices without sacrificing reliability or latency. However, despite the growing demand for URLLC services, problems with resource allocation and network stability exist (Feng et al., 2021).

Mehmeti and La Porta (2021), who put more of an emphasis on particular network management, talked about the problem of admission control for URLLC users in 5G networks. The authors proposed an effective admission control strategy that emphasized the necessity for the best resource allocation to deliver on the promise of high dependability and low latency made by URLLC. The importance of effective resource use is thus highlighted by this work for the real-world application of URLLC (Mehmeti & La Porta, 2021). Popovski et al. (2017) provides a thorough explanation of the tenets and components of URLLC. They discuss the difficulties that ensuring ultra-reliability and low latency entails and suggest solutions to these problems. According to the study, a fundamental change in how wireless communication networks is designed and optimized is necessary to meet the goals of URLLC (Popovski et al., 2017).

In conclusion, URLLC offers a plethora of opportunities for enhancing the capabilities of 5G and beyond networks, but there are still many obstacles to be overcome in terms of resource allocation, network stability, and system design (Feng et al., 2021; Mehmeti & La Porta, 2021; Popovski et al., 2017).

3 Methodology

In this chapter, the methods used to conduct the research are explained. A description of the research methodology, data collecting and analysis procedures, and metrics employed to assess various time synchronization and real-time communication options are also included.

3.1 Research Approach

The methodology of this study aligns with the positivist approach, emphasizing objective, measurable facts. This is appropriate given the focus of this study, which is on clock synchronization LVM networks in a manufacturing setting.

An experimental design was used in this work to enable the manipulation of variables and evaluation of their effects. This entails modifying time synchronization configurations and assessing the effectiveness and precision of those setups. This design decision was influenced by the necessity for accurate measurements and a thorough knowledge of the system's causal relationships.

This research strategy was required by the objectives outlined in the project description. Through a thorough analysis of the state of the art in terms of open-source software, technologies, and research, the study seeks to discover appropriate techniques for time synchronization in an LVM network. The OPC UA TSN, Time-Sensitive Networking (IEEE 802.1 Working Group), and Ultra Reliable Low Latency Communication (URLLC) features of 5G were only a few of the new standards that were made possible by this design. The controlled setting of an experiment allows for measuring and evaluating the impacts of these standards on time synchronization. Moreover, by manipulating different variables associated with each standard, the effects can be isolated and better understood. Finally, the iterative nature of experimental design provides a platform for continuous refinement and optimization based on the findings.

Data for this study was gathered from a variety of sources, including fast signal acquisition with oscilloscope connected to industrial PC performed on signals generated based on another IPC’s clock utilizing the Precision Time Protocol (PTP) output and data logged by a computer using a time synchronization card. A dataset for analysis was produced by the integration of data sources. Multiple tests conducted over long periods of time established the validity and dependability of the findings, enabling extensive data collection and confirmation of consistent results.

In summary, the research methodology employed in this study enabled a detailed examination of time synchronization in LVM networks. The study seeks to offer insightful contributions to discussions on real-time communication in industrial networks through meticulous methodological and ethical issues.

3.2 Data Collection and Analysis

This study's data were collected through an extensive review of academic literature from IEEE Xplore, Springer, and various telecommunication websites and books. Finding the first articles that established each field and then following papers that demonstrated how these areas developed through time and their most recent technological advancements were the main goals of the selection process. Academic verification has been done on each of these papers.

Several keywords associated with real-time communication, time synchronization, LVM networks, and industrial networks were used to search for pertinent material. The terms "PTP", "gPTP", "WhiteRabbit", "EtherCAT", "PowerLink", "ProfiNET IRT", "TSN", "OPC UA TSN", and "URLLC 5G" were a few of them. The literature was manually analyzed by reading through each paper and selecting the ones that provided the most valuable and relevant information.

As a part of this research, two practical experiments were performed. In the first test, an Intel i210 card was used to set up a Precision Time Protocol (PTP) system, and its performance and accuracy were assessed. The PERLE IDS-305 and an 18-port White Rabbit device operated in PTP mode were the two master clocks that were being compared in terms of performance. A digital oscilloscope (RIGOL DS2201A), industrial PCs (ADVANTECH MIC-75M20), a laser detector (THORLABS PDA10A2), and a laser unit (LPS-635FC) were among the tools used in these testing. Digital information was gathered via the oscilloscope and laser testing.

The major goal of these tests was to find a dependable, economical, and precise clock synchronization technique. The outcomes of these experiments should confirm and provide context for the conclusions drawn from the literature review. The results of the oscilloscope and laser tests will be statistically examined, and their performance will be compared. NumPy and pandas are two Python libraries that were used for this investigation.

3.3 Evaluation Metrics

To evaluate the effectiveness of various time synchronization approaches and real-time communication tactics, this research focuses on several important factors. Accuracy, dependability, latency, cost-efficiency, operational consistency over time, performance across a range of devices, and suitability for integration into various machines are among the factors that were selected. Since the main goal of this study is to achieve accurate time synchronization—a fundamental requirement for LVM networks, where even little differences can have significant effects—accuracy is of the utmost significance.

Both reliability and latency are important because industrial systems require unshakable dependability and a short data transmission delay, both of which have a direct impact on the efficiency of the system.

To make sure that the suggested solutions are not unreasonably expensive and can be widely adopted, the cost-efficiency of these strategies is assessed. Additionally, the performance of these systems on various devices and their adaptation to various types of machinery are considered. A software program called ptp4l, which generates several metrics, is largely used to collect measurement data for these factors. The generated information is then examined to determine the performance and stability of the system. Two PCs outfitted with various Intel cards are used for hardware testing in addition to the software-based evaluation. The other functions as the signal, while the first operates as an oscilloscope trigger. The measurement of any potential card errors is possible with this set up. Another test that provides another way to assess performance is coupling a laser to one of the cards and observing the light it emits on an oscilloscope.

This study aims to achieve PTP-level accuracy, which is renowned for its utmost precision. The outcomes are contrasted with three widely used techniques: NTP, PTP, and White

Rabbit. The quality of the devices, the length of the cable, and the GPS signal strength could all have an impact on the outcomes. Therefore, to reproduce this work, one needs a trustworthy GPS clock, the ptp4l program, a trustworthy master clock, and a suitable testing environment. This study assures that the research conducted is complete, trustworthy, and valid by using these criteria and approaches. This thorough method makes it possible to compare various time synchronization and real-time communication strategies in an efficient manner, thereby determining which ones are best for inclusion in multi-party manufacturing networks that use LVM networks.

4 PTP System Implementation and Testing

4.1 PTP System with Intel i210 Card on Linux

4.1.1 Hardware

In these experiments, the Advantech MIC-770 industrial computer, the PERLE IDS-305 managed Ethernet switch, the 18-port White Rabbit device, and the Intel i210 card were all incorporated into the PTP system. Our PTP network architecture is centered around the Intel i210 Ethernet Controller. The card uses a single port and is part of the I210 Series. It offers data transmission at a rate of 1GbE per port. This card's conformance to the IEEE 1588 protocol, which is necessary for the installation of PTP systems, is one of its defining features. But it's important to remember that it does not support the MACsec IEEE 802.1 AE (intel).

In our hardware setup, the Advantech MIC-770 industrial computer acts as a dependable workhorse. The MIC-770 offers a variety of connection choices, including VGA and HDMI for video output and a range of USB ports for peripherals. It uses an Intel® 8th Gen CoreTM I CPU and a Q370/H310 chipset. This device has received acknowledgment in the form of a Microsoft Azure PnP Certification, indicating its consistent performance, and it can resist a wide range of operational temperatures. (28ontinuou)

The PERLE IDS-305 managed Ethernet switch was introduced to play the job of data transmission regulation. This switch provides increased port bandwidth thanks to several useful features like Energy Efficient Ethernet (EEE), 802.3x flow control, reliable TSN capability and Link Aggregation protocol. Furthermore, the network's stability and effectiveness are maintained through its Storm Control and Bandwidth Control Monitoring functions. (perle)

Lastly, the precise time and frequency synchronization capabilities of an 18-port White Rabbit device make it a crucial part of the network. It has been configured specifically to run in PTP mode for this implementation.

4.1.2 Software

A carefully chosen selection of software programs and libraries was used to construct the Precision Time Protocol (PTP) system. These included the operating system Ubuntu 22.04 LTS, the Linux PTP Project’s ptp4l and phc2sys apps, and Python programming augmented by the analytical strength of the Pandas and NumPy libraries. The system was built on top of the open-source operating system Ubuntu 22.04 LTS (Long Term Support). This version guarantees security and maintenance updates for five years, providing ongoing dependability for the PTP system’s software components. The ptp4l and phc2sys applications from the Linux PTP Project were essential to the operation of the system. The PTP clock is implemented in software by the ptp4l program, which also supports boundary clocks. It controls the network interface’s clock synchronization state machine and modifies the system clock to match the PTP network’s master clock. To synchronize the system clock with a PTP hardware clock (PHC) or another system clock, the phc2sys application is crucial. With the help of this adaptable function, the PHC and system clock may be synchronized, ensuring accurate timekeeping between hardware and software components.

Python was used as the programming language for the data analysis component. Python is a mainstay in the field of scientific and numerical computing and is renowned for its simple syntax and extensive capability. Its extensive ecosystem of libraries, particularly the Pandas and NumPy libraries, were utilized. Pandas is a powerful tool for handling and analyzing structured data since it offers fundamental data structures and analytical capabilities. It has a DataFrame object that supports cutting-edge operations like aggregation, merging, and slicing, which are essential for challenging data processing jobs. Contrarily, NumPy provides support for sizable multi-dimensional arrays and matrices in addition to a vast variety of mathematical functions to perform operations on these arrays.

The PTP system data was thoroughly analysed using the capabilities of these libraries. As a result of the thoughtful integration of Ubuntu 22.04 LTS, the ptp4l and phc2sys apps, and the Python libraries Pandas and NumPy, a reliable and effective software environment for the PTP system and analyzing experimental data was made possible. The mechanics of configuring and using these tools will be explained in later sections.

4.1.3 Implementation

The implementation phase began with setting up the Linux operating system, GPS, and master clocks: the PERLE IDS-305 and the 18-port White Rabbit device, both operating in PTP mode. Then, the Intel i210 card was installed on the MIC-770 PC and linked to the master clocks for PTP.

The first command used, *echo 1 > / sys/class/ptp/ptp0/pps_enable*, enabled the Pulse Per Second (PPS) output from the PTP hardware clock (PHC). The PPS signal is key in timekeeping and synchronization as it signifies the passage of time, helping devices align their internal clocks.

Next, the ptp4l application was initiated with the command *sudo ptp4l -2 -E -m -s -I enp1s0*. The flags -2, -E, -m, -s, and -I define various settings. The -2 flag ensures PTP version 2 operation, -E enables Ethernet mode, -m allows detailed protocol state information printing, -s runs ptp4l in “slave” state, and -I enp1so specifies the network interface. The output from ptp4l provides critical timing information for synchronization. The system’s direct output was recorded, and data was subsequently gathered, processed, and analyzed using Python, NumPy, and Pandas. These libraries made it easier to evaluate the findings in-depth and to spot trends in the data.

Despite careful setup, some lab-related issues led to anomalies in the output, an unusually large numbers for a few seconds. Certain values were not included in the analysis and were removed from the original data. After careful testing and validation, the PTP system’s output was compared to established benchmarks to verify correct synchronization and efficient PTP system operation.

4.2 Testing Methodologies

4.2.1 Oscilloscope Testing

A RIGOL DS2201A Digital Oscilloscope was used for the oscilloscope test. Cables attached into channels 1 and 2 were used to connect two outputs from the cards to the oscilloscope;

one of the cables was also connected to the trigger pin. We used two commands that were built within the cards to generate pulses. Both commands were intended to generate pulses, one at a frequency of 1 hertz and the other at 1 megahertz. The respective commands were as follows:

```
$ sudo ./testptp -d /dev/ptp0 -L 0,2 -I 0
```

The 'testptp' utility Is a tool designed for Linux that aids interaction with PTP hardware clocks. The ‘-d /dev/ptp0’ command specifies the device being used, in this case, the first PTP device. The ‘-L 0,2’ argument is used to set the clock’s time according to the system time and to initiate output signals at the frequency specified by the ‘-p’ argument in the next commands. The ‘-I 0’ argument designates the output signal channel to be used.

```
$ sudo ./testptp -d /dev/ptp0 -p 1000000000 -I 0
```

```
$ sudo ./testptp -d /dev/ptp0 -p 1000 -I 0
```

The ‘-p 1000000000’ and ‘-p 1000’ commands set the frequency of the output signal. These commands, therefore, direct the card to generate signals at frequencies of 1 megahertz and 1 hertz, respectively. Both cards were plugged into the White Rabbit device with ethernet cable as boundary clocks and were in PTP mode for the initial test. In the second test, the White Rabbit device was attached to the trigger, and the PERLE was connected to the other card. The configuration for the third test was different from the second test in that the trigger was connected to the PERLE device, while the other card was connected to the White Rabbit device. Finally, both cards connected to the PERLE in the fourth test.

The oscilloscope's data was 31ontinuoussly tracked and saved on a computer throughout each test. The oscilloscope’s programming was one of the difficulties. Nevertheless, despite these challenges, a substantial data set was compiled. This information was compiled into a.txt file and had about 1400 data points on each second. Key observations were made at the points where the signal’s value exhibited a sharp change. The most important information about signal synchronization was provided by these places.

The gathered dat’ from this oscilloscope test supplied crucial insights into the operation and efficiency of the PTP system. Chapter 5 will offer a thorough analysis and explanation of these findings.

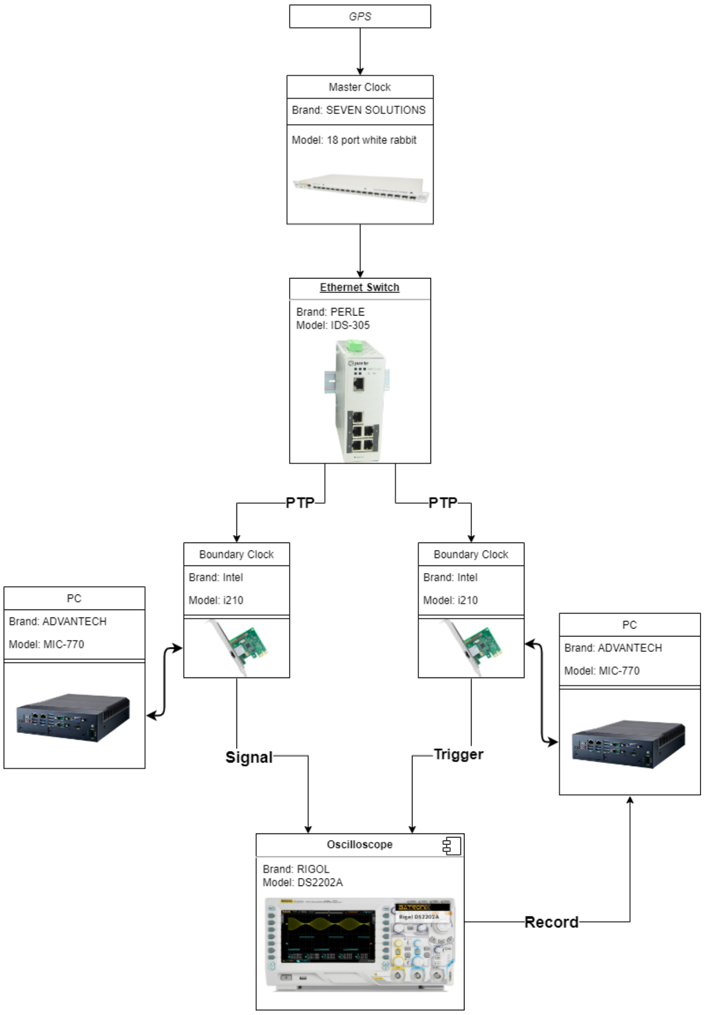

Figure 2: oscilloscope test diagram

4.2.2 Laser Testing

In the laser testing, a similar setup to the oscilloscope test was used, but with the addition of a laser and an optical detector. The reason for changing an electrical signal route to optical was to simulate an optical measurement made by a computer-controlled device in an industrial environment. The Intel i210 card was connected to the laser via a buffer amplifier because the i210 programmable I/O pins can drive only very small amount of current, while the laser detector was linked to the channel 1 input. A 10 ohm resistor was incorporated into the connection between the laser and the card. This, along with maintaining a steady 50 milliamps current draw by the laser, ensured optimal operation. The laser beam was collimated by a lens at the end of the optical fibre. The detector was positioned to align with the laser beam's path.

Like the oscilloscope test, the following commands were executed:

```
sudo ./testptp -d /dev/ptp0 -p 1000000000 -I 0

sudo ./testptp -d /dev/ptp0 -p 1000 -I 0
```

Following this, four testing configurations were implemented. In the first, both the pulse-generating and pulse-detecting cards were linked to the White Rabbit device. This tested the White Rabbit's performance and the system's synchronization capabilities.

In the second setup, the Intel i210 card was connected to the White Rabbit, and the trigger card was linked to the Perle IDS-305. This arrangement tested the system's synchronization when different devices were used for pulse generation and detection.

In the third configuration, the setup was flipped. The trigger card was connected to the White Rabbit, while the Intel i210 card was tied to the Perle IDS-305. This offered another perspective on the system's performance under varying setups.

Lastly, both the Intel i210 card and the trigger card were linked directly to the Perle IDS-305. This configuration examined the Perle IDS-305's performance and the system's synchronization capabilities under this setup.

These configurations provided insights into system performance, the performance of the White Rabbit and Perle IDS-305, and how device choice and card positioning influenced the system's time synchronization accuracy. Further details on this data analysis will be presented in Chapter 5.

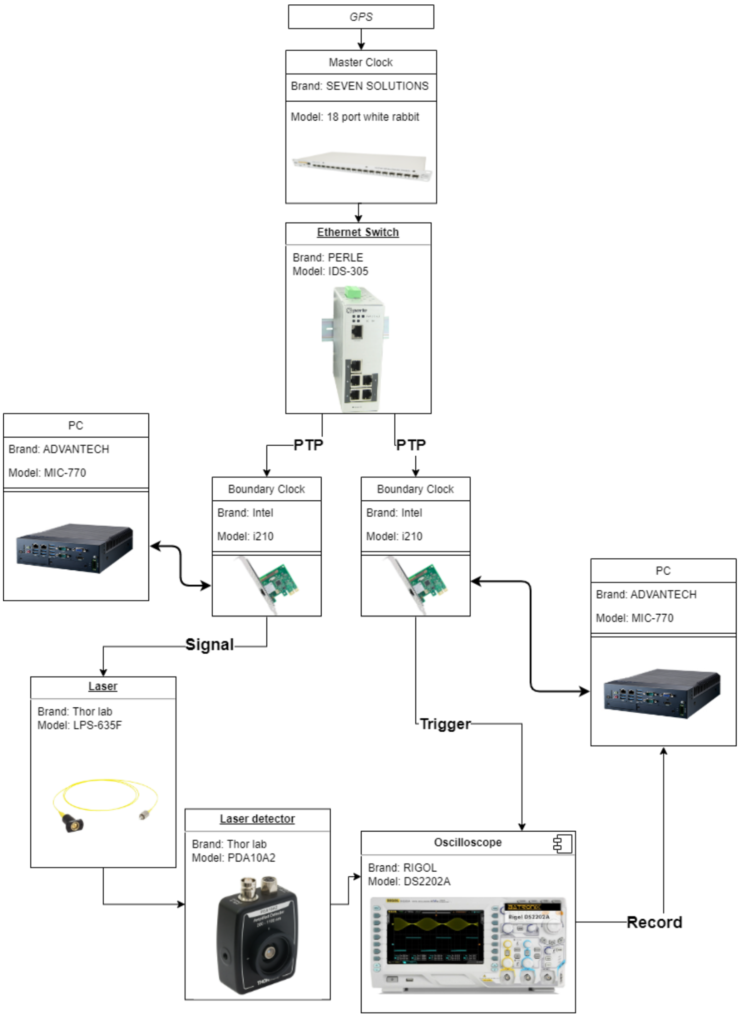

Figure 3: Laser test diagram

5 Results and Discussion

The findings from the oscilloscope and laser experiments are presented in Chapter 5, accompanied by an understandable analysis of these results. Furthermore, the potential implications of these findings for the addition of time synchronization and real-time communication to LVM networks are also discussed.
The selected method for outlier detection in this dataset is the Interquartile Range (IQR) method. Developed by Tukey (1977), this technique provides an efficient and robust way of identifying outliers in a dataset, without making assumptions about the data distribution. The IQR, as a measure of statistical dispersion, quantifies the difference between the upper (Q3) and lower quartiles (Q1) of the dataset. In the context of outlier detection, outliers are defined as any data points that fall below Q1 - 1.5IQR or above Q3 + 1.5IQR. These boundaries create a range where most of the data, excluding outliers, are expected to lie.

This method is particularly suited to the dataset in consideration due to its robustness and simplicity. With the dataset consisting of integer values ranging from -2500 to 2500, and with a presence of very few outliers, the IQR method can effectively determine these extreme values without being overly influenced by them. Unlike methods that depend on mean and standard deviation, such as the Z-score method, the IQR method is less sensitive to extreme values. This sensitivity of mean-based methods could potentially skew the results and incorrectly identify outliers if the extreme values are particularly pronounced (Hubert & Vandervieren, 2008).

Another key advantage of the IQR method is its lack of dependence on the distribution of the data. It does not require the data to follow a specific distribution, such as a normal distribution, which is a common assumption for many statistical methods. This makes the IQR method a versatile and reliable choice for outlier detection in various kinds of datasets, including the one in question. Furthermore, the IQR method is computationally efficient, which is an important consideration given the size of the dataset, making it an appropriate choice for this application.

5.1 ptp4l Output Test Result

The first 12 line of the output of the ptp4l program is as follows:

```
ptp4l[2595537.059]: selected /dev/ptp0 as PTP clock
ptp4l[2595537.116]: port 1: INITIALIZING to LISTENING on INIT_COMPLETE
ptp4l[2595537.116]: port 0: INITIALIZING to LISTENING on INIT_COMPLETE
ptp4l[2595538.849]: port 1: new foreign master 64fb81.fffe.2ffba4-17
ptp4l[2595542.697]: selected best master clock 001bc5.fffe.090844
ptp4l[2595542.697]: updating UTC offset to 37
ptp4l[2595542.697]: port 1: LISTENING to UNCALIBRATED on RS_SLAVE
ptp4l[2595545.221]: master offset        185 s0 freq  -24795 path delay       237
ptp4l[2595546.162]: master offset        186 s2 freq  -24794 path delay       237
ptp4l[2595546.162]: port 1: UNCALIBRATED to SLAVE on MASTER_CLOCK_SELECTED
ptp4l[2595547.010]: master offset        194 s2 freq  -24600 path delay       240
ptp4l[2595547.889]: master offset         19 s2 freq  -24717 path delay       244
```

The output from the ptp4l program provides insightful data about time synchronization. Here's an interpretation of what was seen:

"selected /dev/ptp0 as PTP clock": The program had chosen a clock for management.

"port 1: INITIALIZING to LISTENING on INIT_COMPLETE" and "port 0: INITIALIZING to LISTENING on INIT_COMPLETE": It was confirmed that the startup process was completed successfully.

The system then started to identify other clocks in the network:

"port 1: new foreign master 64fb81.fffe.2ffba4-17": Another clock in the network was found.

"selected best master clock 001bc5.fffe.090844": The best clock to synchronize with was selected.

"updating UTC offset to 37": An adjustment was made for the difference in time due to leap seconds.

Next, synchronization with the master clock was attempted:

"port 1: LISTENING to UNCALIBRATED on RS_SLAVE": The process of matching the master clock's time was initiated.

"port 1: UNCALIBRATED to SLAVE on MASTER_CLOCK_SELECTED": Calibration was completed, and synchronization with the master clock was begun.

The following lines provided an evaluation of the synchronization:

"master offset xxx s0/s2 freq yyyyy path delay zzz": This information is like a report on the time synchronization process. "master offset" indicates how different the system clock is from the master, "freq" shows how quickly the system clock is adjusting to match the master, and "path delay" tells how long messages take to travel from the master clock to the system clock.

Here are the results of these experiments:

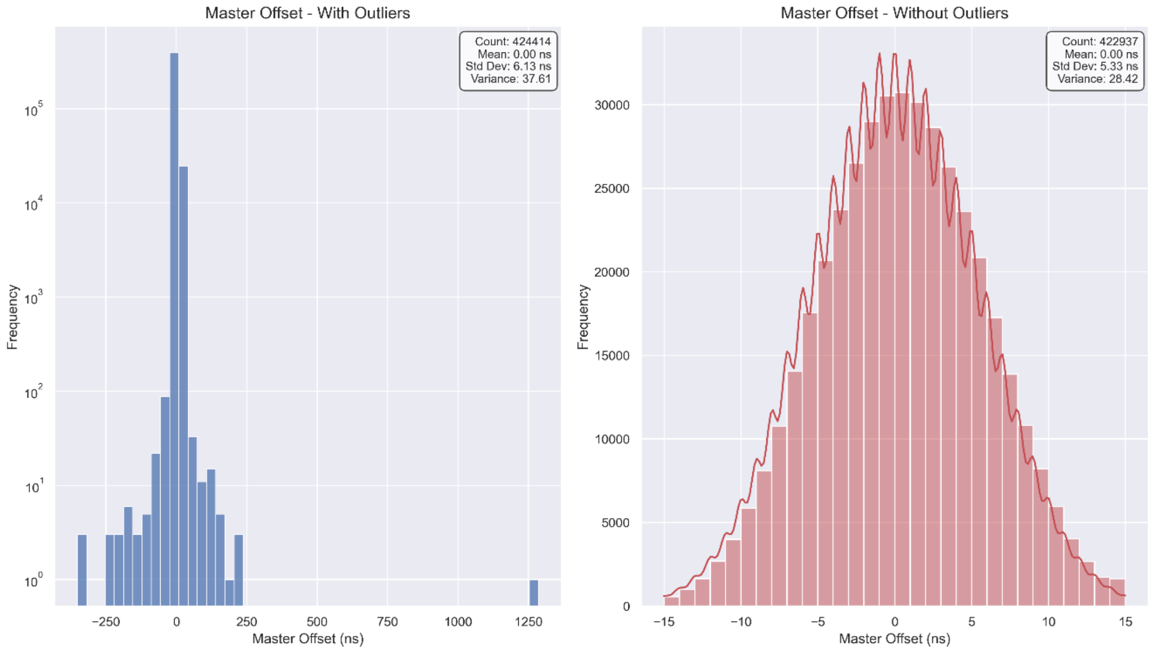

Figure 4: ptp4l output of the white rabbit device in PTP mode

Note that the line on the graphs is their KDE. Kernel Density Estimation (KDE) is a non-parametric method employed to estimate the probability density function of a given set of data. The technique, based on statistical theory, revolves around the usage of a kernel, which is a smooth and peaked function. The underlying idea is to superimpose this kernel function over each data point in the dataset, sum them all together, and normalize them to construct a smooth approximation of the dataset's distribution. The choice of kernel and its corresponding bandwidth significantly influence the resulting estimation, with common options for the kernel including Gaussian, Epanechnikov, and Exponential functions among others.

In the context of the given data visualization task, KDE is utilized to provide a more seamless and comprehensive depiction of the data distribution. The default histogram, while effective in illustrating the raw frequency of data points in pre-specified bins, often presents a choppy and disconnected view of the data. This can sometimes obscure the overall shape and trends of the distribution. By employing KDE, the distribution of the dataset is portrayed in a smooth, continuous form that enhances comprehension of the data's underlying structure. Notably, the KDE plot reveals more nuanced information about the distribution, including the presence of multiple modes, skewness, or subtle variations in density that may not be evident in the traditional histogram. Furthermore, using KDE in tandem with filtering techniques, such as the Interquartile Range (IQR) method for outlier removal, permits an even more refined exploration and representation of the 'typical' data behavior, further enhancing the usefulness of the KDE approach in this context.

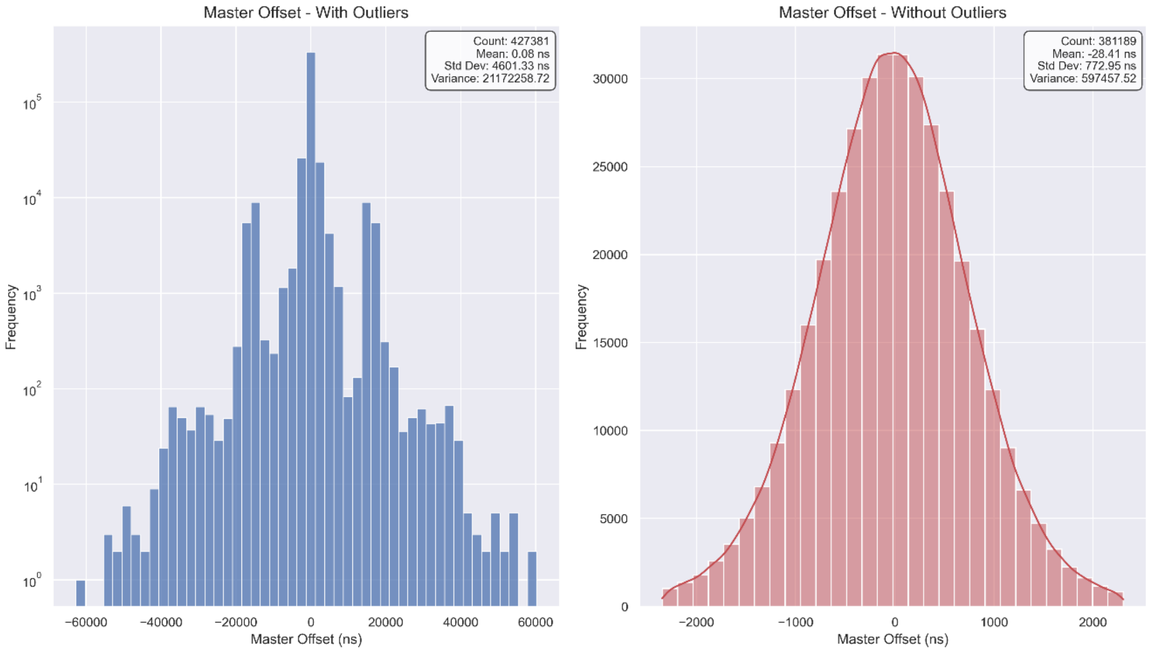

Figure 5: ptp4l output of the perle IDS-305

The plot on the left has a logarithmic y axis.

These results enable an assessment of the effectiveness of time synchronization and help in identifying potential improvements.

5.2 Oscilloscope Test Results

Four experiments were conducted in this section. First using the White rabbit device on PTP mode which was expected to have the best results as it did.

It should be noted that the first plots with the unfiltered data have logarithmic y axis.

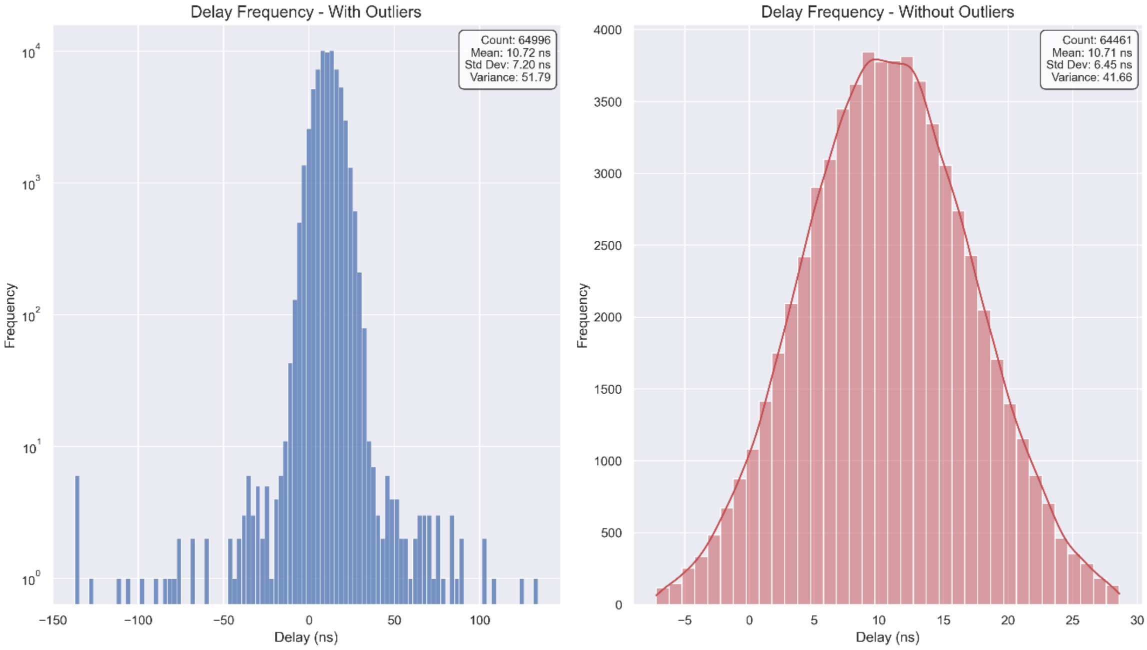

Figure 6: Laser test with white rabbit device in PTP mode

The next two experiments were conducted with one of the intel cards connected to the perle IDS device:

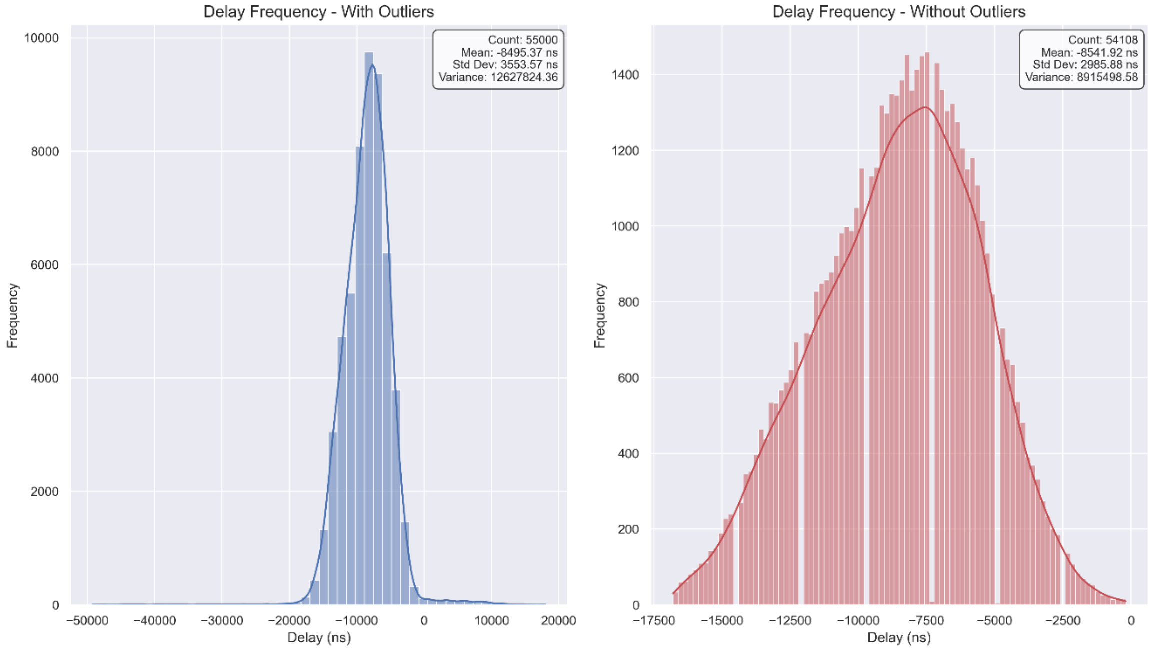

Figure 7: Trigger signal connected to perle

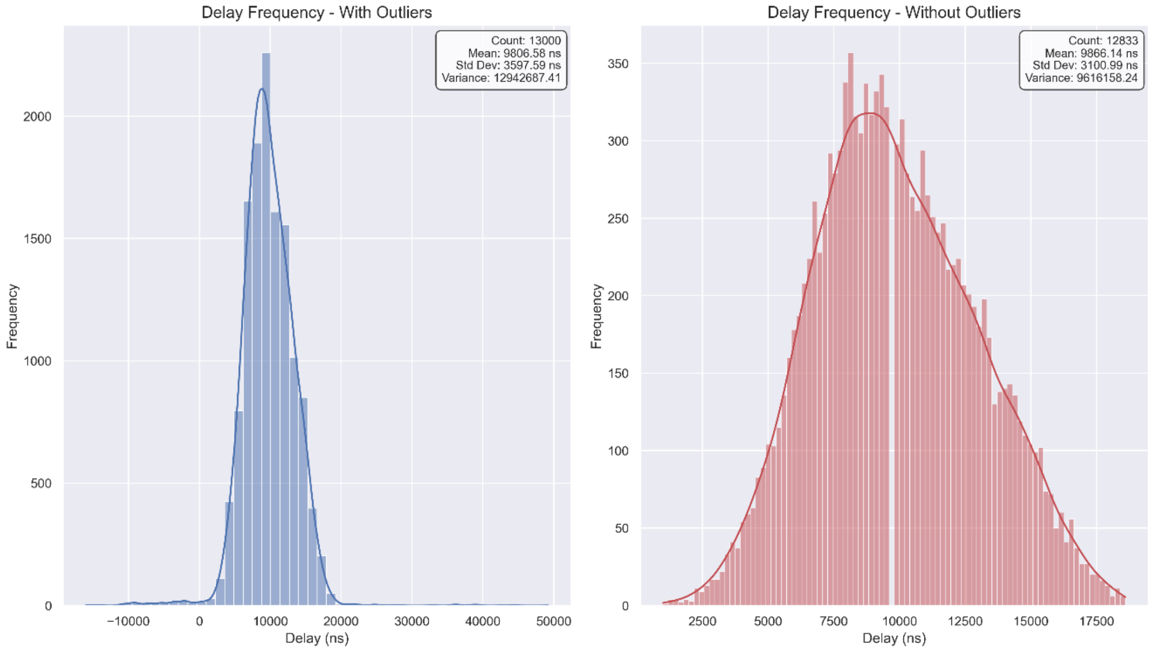

Figure 8: Measured signal of the card connected to perle

And lastly the test of both cards connected to perle master clock:

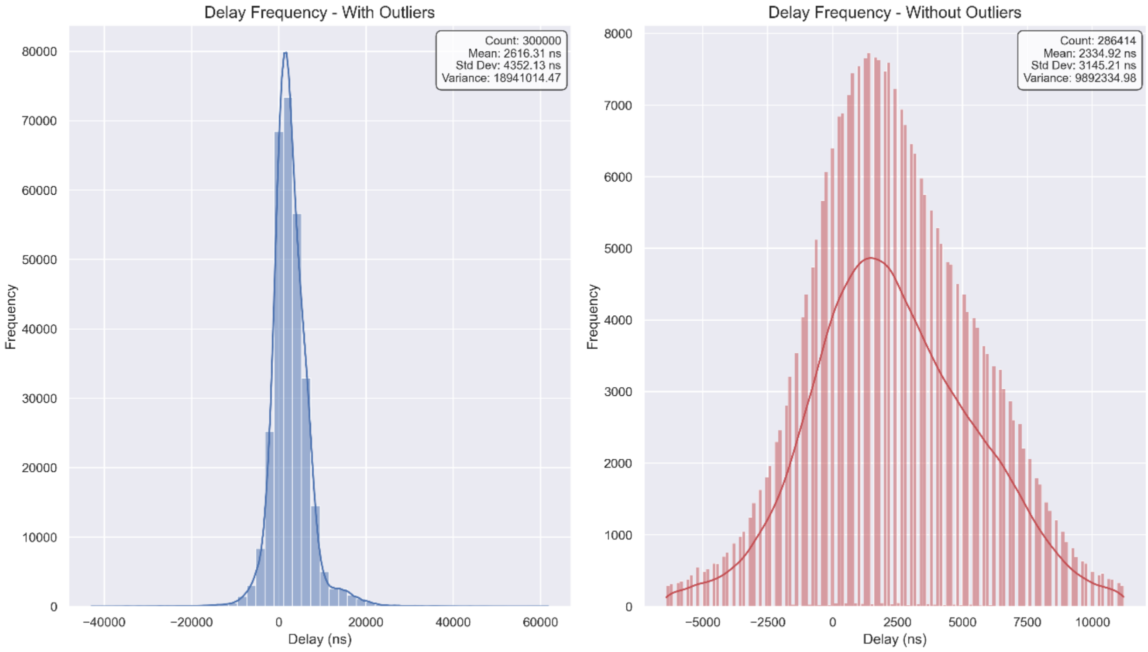

Figure 9: Both cards connected to perle

Insights were gained into the performance of the PERLE IDS-305 Managed Ethernet Switch through the laser test results. The focus was on how the device influenced signal accuracy and delay.

The first thing to look at is how the results were presented in a graph. It's notable that a logarithmic y-axis was used for the first plot. This was due to some very high values which would have made the plot hard to read on a regular axis. By using a logarithmic scale, a clear and complete view of all results could be seen.

Next, it was observed that the accuracy of the results was reduced significantly when the PERLE IDS-305 was used. This could be due to the clocking mechanism inside the PERLE IDS-305. It might not be as precise or consistent as those found in more specialized time synchronization devices. This might cause slight inconsistencies or delays in how times are stamped and transmitted, leading to the higher average delay and reduced precision seen in the tests. Another factor could be the time the PERLE IDS-305 takes to process and forward packets, which can also introduce extra delays. When the PERLE IDS-305 was used as the signal source, there was a big increase in the average delay compared to the other setups. But when it was the trigger, the average delay was the smallest, even reaching a negative value.

A test was also conducted with just one card connected to the PERLE IDS-305. This was to show the effect of these changes. However, such a setup would not be practical in real-world use.

The laser test results suggest that the PERLE IDS-305 may be a good choice for applications that need accuracy on a microsecond scale. But for tasks requiring precision at the nanosecond level, a more sophisticated and probably more expensive device would likely be needed.

5.3 Laser Test Results

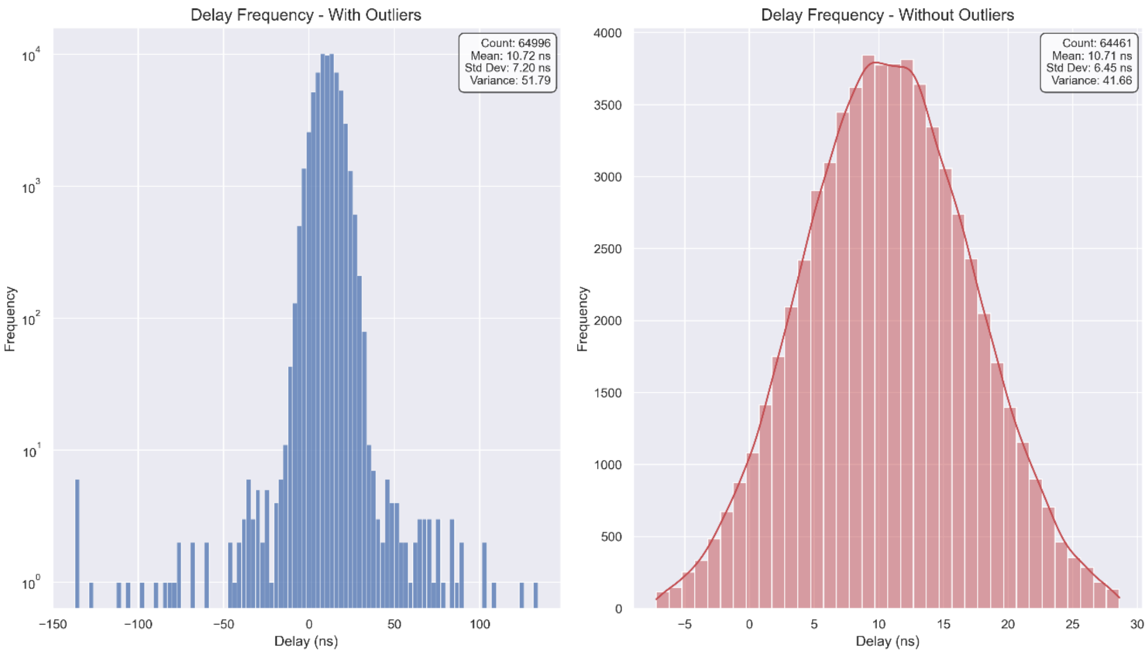

Figure 10: Laser test with White rabbit device in PTP mode master clock

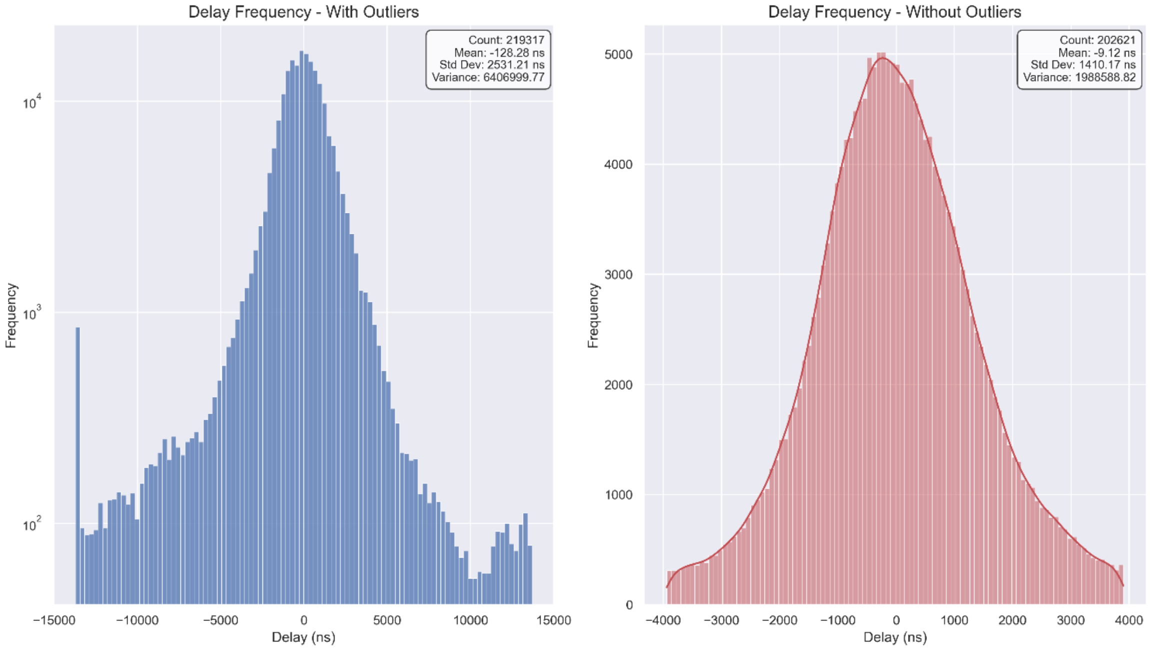

Figure 11: Laser test with perle master clock

The laser tests provided useful information about how well the system performed under different setups. Both the White Rabbit device and the PERLE IDS-305 Managed Ethernet Switch played important roles in these results. The White Rabbit device showed much better results in terms of standard deviation (std) and mean delay. This means it provided a consistent level of time synchronization. This is likely because the White Rabbit device is specially designed to be precise and consistent, which is exactly what's needed for advanced industrial or scientific uses.

The performance of the PERLE IDS-305 was not as good as the White Rabbit's in terms of precision. However, its average delay wasn't too bad, especially compared to the results from the oscilloscope tests. The reason the PERLE IDS-305 did better in the laser tests might be due to the different ways these two tests measure time synchronization accuracy. Each test has its own conditions and measures, which could affect how well the devices perform. It's also important to remember that the PERLE IDS-305 Managed Ethernet Switch is mainly made for general industrial use, not tasks that need very precise time synchronization. So, the fact that it had a decent average delay could be seen as a sign of its strength and flexibility in different situations. It might be a good choice for tasks where accuracy at the microsecond level is good enough.

5.4 Comparison and Analysis of Test Results

Name of the test	Average delay (ns)	Stdv of delay (ns)
WR_ptp4l	0.00	5.33
Perle_ptp4l	-28.41	772
WR_osc	6.55	3.08
Perle_osc_signal	9866	3101
Perle_osc_trigger	-8542	2986
Perle_osc_both	2335	3145
WR_laser	10.71	6.45
Perle_laser	-9.12	1410

WR_ptp4l vs Perle_ptp4l: WR_ptp4l shows superior precision and consistency, with an average delay close to zero and a very small standard deviation. On the other hand, Perle_ptp4l shows a slightly negative average delay and a much larger standard deviation, implying lower precision and consistency.

WR_osc vs Perle_osc (signal, trigger, both): WR_osc demonstrates a small average delay and very small standard deviation, suggesting good precision and stability. In contrast, all Perle_osc tests show larger average delays and much higher standard deviations, indicating less precision and stability. The "signal" test shows a significant positive delay, the "trigger" test a significant negative delay, as a sign of tiggering happening before the signal, and the "both" test a moderate positive delay, suggesting potential inaccuracies in different aspects of the synchronization process.

WR_laser vs Perle_laser: Both show small average delays, with WR_laser being positive and Perle_laser being negative. However, WR_laser is more consistent, as indicated by a much smaller standard deviation.

The higher cost of the White Rabbit (WR) setup might be justified given its superior performance, as indicated by smaller average delays and smaller standard deviations in the tests performed. This suggests higher precision and stability, which could be crucial for systems requiring precise time synchronization in ns precision level.

However, the decision should also take into account the specific requirements of the application. If the system can tolerate slightly higher delay variations and the cost difference is a significant factor, the Perle setup could be a more cost-effective solution.

So, while the WR setup does show better performance in these tests, it's important to balance that against its significantly higher cost. A careful cost-benefit analysis, considering not only the time synchronization requirements but also budget constraints, will help in deciding the better solution for your specific case.

6 Conclusions

In this section, a synthesis of the primary findings from the research is provided, placing the results within the broader context of time synchronization and real-time communication in large networks.

Valuable insights into the process of time synchronization in industrial networks were gleaned from the thorough interpretation of the ptp4l program output. The examination of these results contributes significantly to the literature on time synchronization processes. The performance of both the White Rabbit and Perle IDS-305 setups was assessed through the execution of oscilloscope and laser tests. The findings mirrored the expected performance attributes of these devices: the White Rabbit setup in PTP mode exhibited superior precision, while the Perle IDS-305, despite being designed for general industrial use, demonstrated a commendable performance.

From a practical perspective, this research makes an important contribution by enhancing the understanding of time synchronization in a network, thus informing more effective decision-making processes when choosing between different setup options based on application requirements and cost-effectiveness. On a societal level, the research encourages the pursuit of precision and efficiency within industrial settings, potentially leading to cost savings and increased productivity. Despite these valuable findings, certain limitations were inherent within the scope of this research. The devices selected for this study, namely the White Rabbit and Perle IDS-305, may restrict the extrapolation of the findings to other devices or setups. Additionally, the results are contingent upon the specific tests conducted and may vary under different testing conditions or parameters.

Suggestions for future research include an exploration of alternative time synchronization methods and devices, as well as testing under a more diverse range of conditions. Efforts could also be directed towards optimizing time synchronization within large networks, focusing particularly on enhancing precision while minimizing cost. This would widen the potential for high-precision time synchronization applications across a variety of industries.

References

Carmignato, S., Savio, A. & Savio, E., 2013. Advances in Large-Scale Metrology – Review and future trends.

Gore, R. N., Lisova, E., Åkerberg, J. and Björkman, M., 2020. Clock Synchronization in Future Industrial Networks: Applications, Challenges, and Directions. AEIT International Annual Conference, Catania, Italy, pp. 1-6.

Han, S., Zhu, X., Mok, A. K., Chen, D. and Nixon, M., 2011. Reliable and Real-Time Communication in Industrial Wireless Mesh Networks. 17th IEEE Real-Time and Embedded Technology and Applications Symposium, Chicago, IL, USA, pp. 3-12.

Hockenberry, J., Song, J. & Stone, J., 2015. Laser Trackers for Large Scale Dimensional Metrology: A Review.

Matsakis, D., Levine, J., & Lombardi, M. A. (2018). Metrological and legal traceability of time signals. United States Naval Observatory, Washington, DC, USA, and Time and Frequency Division, National Institute of Standards and Technology, Boulder, Colorado, USA.

Carmignato, S., Savio, A., & Savio, E. (2013). Advances in Large-Scale Metrology – Review and future trends.

Mills, D. L. (1991). Internet time synchronization: the network time protocol. IEEE Transactions on communications, 39(10), 1482-1493.

Eidson, J. C. (2006). Measurement, control, and communication using IEEE 1588. Springer Science & Business Media.

Bregni, S. (2002). Synchronization of digital telecommunications networks. John Wiley & Sons, Ltd.

Agustoni, M., Castello, P., Frigo, G., & Gallus, G. (2022). Time Synchronization Sensitivity in SV-based PMU Consistency Assessment.

Gore, R. N., Lisova, E., Åkerberg, J., & Björkman, M., 2020. Clock Synchronization in Future Industrial Networks: Applications, Challenges, and Directions. AEIT International

Annual Conference (AEIT), Catania, Italy, 2020, pp. 1-6, doi: 10.23919/AEIT50178.2020.9241154.

Thakur, A., & Satish, M., 2022. Clock Synchronization in Distributed Systems. Available at: https://www.researchgate.net/publication/359685826_Clock_Synchronization_in_Distributed_Systems.

IEEE, 2023. IEEE 1588 for Clock Synchronization in Industrial IoT and Related Applications: A Review on Contributing Technologies, Protocols and Enhancement Methodologies.

UKEssays, 2018. Importance Of Time in Distributed Systems. Available at: https://www.ukessays.com/essays/philosophy/importance-of-time-in-distributed-systems-philosophy-essay.php?vref=1.

Sampath, A., & Tripti, C., 2010. Synchronization in Distributed Systems. In 2010 International Conference on Information Networking and Automation (pp. 1-6). IEEE.

Chen, G., Hao, W., & Xiaofeng, T. (2015). Clock synchronization technology based on FPGA. 2015 IEEE International Conference on Communication Software and Networks (ICCSN).

Mills, D.L. (1991). Internet Time Synchronization: The Network Time Protocol. Electrical Engineering Department, University of Delaware.

Mills, D.L. (1994). Precision Synchronization of Computer Network Clocks. Electrical Engineering Department, University of Delaware.

Rangaswamy, S., & Murthy, S. (2021). An Overview of Network Time Protocol. High Technology Letters, 27(6), 1-8.

Idrees, Z., Ahmad, R., & Khan, A. (2020). IEEE 1588 for Clock Synchronization in Industrial IoT and Related Applications: A Review on Contributing Technologies, Protocols and Enhancement Methodologies. IEEE Access, 8, 155660-155678.

Rezabek, F., Helm, M., Leonhardt, T., & Carle, G. (n.d.). PTP Security Measures and their Impact on Synchronization Accuracy. I8 Network Architectures and Services, Technische Universitat M ¨ unchen ¨ Germany.

Wiesner, A., & Kovácsházy, T. (2021). Portable, PTP-based Clock Synchronization Implementation for Microcontroller-based Systems and its Performance Evaluation. In 2021 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication (ISPCS), NA, FL, USA (pp. 1-6).

Yuan Kexin & Guo Xuebing & Tian Jianying (2021). Research and Implementation of Clock Synchronization Technology Based on PTP. Journal

Fedullo, T., Morato, A., Tramarin, F., Rovati, L. and Vitturi, S., 2022. A Comprehensive Review on Time Sensitive Networks with a Special Focus on Its Applicability to Industrial Smart and Distributed Measurement Systems. Sensors, [online] 22(4), p.1638. Available at: https://doi.org/10.3390/s22041638 [Accessed 20 May 2023].

Lázaro, J., Cabrejas, J., Zuloaga, A., Muguira, L. and Jiménez, J., 2022. Time Sensitive Networking Protocol Implementation for Linux End Equipment. Technologies, [online] 10, p.55. Available at: https://doi.org/10.3390/technologies10030055 [Accessed 20 May 2023].

IEEE 802.1AS-2011, 2011. Standard for Local and Metropolitan Area Networks—Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks. [online] Available at: https://ieeexplore.ieee.org/document/5741898 [Accessed 20 May 2023].

Rezabek, F., Bosk, M., Paul, T., and others, 2022. EnGINE: Flexible Research Infrastructure for Reliable and Scalable Time Sensitive Networks. Journal of Network and Systems Management, [online] 30, p.74. Available at: https://doi.org/10.1007/s10922-022-09686-0 [Accessed 20 May 2023].

Yang, Z., Ma, Y., Yang, W., & Zhang, Y. (2022). Implementation of White Rabbit Time Synchronization System in State Acquisition System of High-energy Physics Experimental Device. Journal of Physics: Conference Series, 2264, 012014.

Hennig, W., & Hoover, S. (2021). White Rabbit Time Synchronization for Radiation Detector Readout Electronics. United States.

Neelam, M. P., Olaniya, H., Rathore, L., Sharma, A., Roy, S., De, S., & Panja, S. (2021). Precise Time Synchronization and Clock Comparison Through a White Rabbit Network-Based Optical Fiber Link.

Rizzi, M., Lipinski, M., Ferrari, P., Rinaldi, S., & Flammini, A. (2018). White Rabbit Clock Synchronization: Ultimate Limits on Close-In Phase Noise and Short-Term Stability Due to FPGA Implementation. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, 65(9), 1726-1737.

Rangaswamy, S., & Murthy, S. (2021). An Overview of Network Time Protocol. High Technology Letters, 27(6), 1-8.

Mills, D.L. (1991). Internet Time Synchronization: The Network Time Protocol. Electrical Engineering Department, University of Delaware.

Han, S., Zhu, X., Mok, A. K., Chen, D., & Nixon, M. (2011). Reliable and Real-Time Communication in Industrial Wireless Mesh Networks. 2011 17th IEEE Real-Time and Embedded Technology and Applications Symposium, Chicago, IL, USA, pp. 3-12.

Ginthör, D., Guillaume, R., Nayak, N., & von Hoyningen-Huene, J. (2021). Time-Sensitive Networking for Industrial Control Networks. In: Mahmood, N.H., Marchenko, N., Gidlund, M., & Popovski, P. (eds) Wireless Networks and Industrial IoT. Springer, Cham.

Sundaresan, S. & Bettati, R. (1997). Distributed Connection Management for Real-Time Communication over Wormhole-Routed Networks.

Doyle, P. (2004). Introduction to Real-Time Ethernet 1. The Extension, 5, 4 – via Contemporary Control Systems, Inc.

Hibbard, J. (2016) 5 real-time, ethernet-based Fieldbuses compared, Automate. Available at: https://www.automate.org/tech-papers/5-real-time-ethernet-based-fieldbuses-compared (Accessed: 21 May 2023).

Eugster, P. T., Felber, P. A., Guerraoui, R., & Kermarrec, A.-M. (2003). The many faces of publish/subscribe. ACM Computing Surveys, 35(2), 114–131.

Industrial Internet Consortium. (2018). Time Sensitive Networks for Flexible Manufacturing Testbed - Description of Converged Traffic Types.

Jasperneite, J., Schumacher, M., & Weber, K. (2007). Limits of increasing the performance of industrial Ethernet protocols. In Emerging Technologies and Factory Automation, IEEE Conference on (pp. 17–24).

Pfrommer, J., Ebner, A., Ravikumar, S., & Karunakaran, B. (2018). Open Source OPC UA PubSub over TSN for Realtime Industrial Communication. ETFA 2018. https://doi.org/10.1109/ETFA.2018.8502479.

Schriegel, S., & Jasperneite, J. (2007). Investigation of industrial environmental influences on clock sources and their effect on the synchronization accuracy of IEEE 1588. In Precision Clock Synchronization for Measurement, Control, and Communication, IEEE International Symposium on (pp. 50–55).

Damm, M., Leitner, S.-H., & Mahnke, W. (2009). OPC Unified Architecture. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-68899-0

González, I., Calderón, A. J., Figueiredo, J., & Sousa, J. (2019). A Literature Survey on Open Platform Communications (OPC) Applied to Advanced Industrial Environments. Electronics, 8(5), 510. https://doi.org/10.3390/electronics8050510

Mahnke, W., Leitner, S.-H., & Damm, M. (2010). OPC UA security features. In Proceedings of the 2010 8th IEEE International Conference on Industrial Informatics, INDIN 2010 (pp. 443–448). IEEE. https://doi.org/10.1109/

Intel. Intel® Ethernet Controller I210 Series. Available at: https://ark.intel.com (Accessed: 25 May 2023).

Advantech MIC-770 - Intel® 8th Gen Core™ i. Available at: https://www.advantech.com (Accessed: 25 May 2023).

Perle IDS-305 industrial managed ethernet switch. Available at: https://www.perle.com/products/switches/ids-305-industrial-managed-ethernet-switch.shtml

Hubert, M. and Vandervieren, E., 2008. An adjusted boxplot for skewed distributions. Computational Statistics & Data Analysis, 52(12), pp.5186-5201.

Tukey, J. W., 1977. Exploratory Data Analysis. Addison-Wesley.