REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

FPGA IMPLEMENTATION OF IEEE 1588 PRECISION TIME PROTOCOL FOR AVIONICS SYSTEMS

Muhammed Emin YABACIOĞLU

MASTER OF SCIENCE THESIS
Department of Avionics Engineering
Program of Avionics Engineering

Supervisor
Prof. Dr. Şeref Naci ENGİN

Co-supervisor
Dr. İbrahim HÖKELEK

February, 2022

REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

FPGA IMPLEMENTATION OF IEEE 1588 PRECISION TIME PROTOCOL FOR AVIONICS SYSTEMS

A thesis submitted by Muhammed Emin YABACIOĞLU in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE** is approved by the committee on 16.02.2022 in Department of Avionics Engineering, Program of Avionics Engineering.

Prof. Dr. Şeref Naci ENGİN
Yıldız Technical University
Supervisor

Dr. İbrahim HÖKELEK
TÜBİTAK BİLGEM
Co-supervisor

Approved By the Examining Committee

Prof. Dr. Şeref Naci ENGİN, Supervisor
Yıldız Technical University _____

Asst. Prof. Onur AKBATI, Member
Yıldız Technical University _____

Asst. Prof. Selçuk CEVHER, Member
Karadeniz Technical University _____

I hereby declare that I have obtained the required legal permissions during data collection and exploitation procedures, that I have made the in-text citations and cited the references properly, that I haven't falsified and/or fabricated research data and results of the study and that I have abided by the principles of the scientific research and ethics during my Thesis Study under the title of FPGA Implementation of IEEE 1588 Precision Time Protocol for Avionics Systems supervised by my supervisor, Prof. Dr. Şeref Naci ENGİN. In the case of a discovery of false statement, I am to acknowledge any legal consequence.

Muhammed Emin YABACIOĞLU

Signature

Dedicated to my family

ACKNOWLEDGEMENTS

I would like to express my appreciation to my advisor, Şeref Naci Engin, for his support of this thesis. I want to express my gratitude to my co-advisor, İbrahim Hökelek, who directed me throughout the research and contributed the most to its completion, as well as Muhammet Selim Demir, who never hesitated to offer guidance and support. I am confident that what I learned from them will be valuable to me in the future.

In addition to all of this, I want to express my gratitude to my precious wife and daughter for their continuous moral support and patience as I concentrated on the thesis. Also, I would like to show my thankfulness to my entire family, particularly my mother and father, who have worked hard to support me come to this days.

Muhammed Emin YABACIOĞLU

TABLE OF CONTENTS

LIST OF SYMBOLS

Δ_{CP}	Clock Pulse Difference between Master and Slave
$CP_{variation}$	Clock Pulse Variation
β_{SI}	Counter Value Recorded by Slave Node
t_3	Delay Request Packet Transmitting Time from Slave
t_4	Delay Request Packet Transmitting Time to Master
Δ_{MS}	Master to Slave Path Delay
µs	Microsecond
f_{clk}	Node Operation Frequency
$T_{operating}$	Node Operation Period
α_{SI}	Number of Pulses
γ_{CI}	Rate Correction Period
N	Sample Count
Δ_{SM}	Slave to Master Path Delay
T_{SI}	Sync Packet Interval
t_2	Sync Packet Receiving Time to slave
t_1	Sync Packet Transmitting Time from Master
Δ_T	Time Deviation of Node
Δt	Time Offset between Master and Slave

LIST OF ABBREVIATIONS

AFDX	Avionics Full-Duplex Switched Ethernet
AM	Amplitude Modulation
ARINC	Aeronautical Research Incorporated
BMCA	Best Master Clock Algorithm
CERN	Conseil Européen pour la Recherche Nucléaire
CPU	Central Processing Unit
DC	Direct Current
E2E	End to End
EMI	Electromagnetic Interference
ES	End System
FCS	Frame Check Sequence
FPGA	Field Programmable Gate Array
Gpbs	Gigabit per Second
GPS	Global Positioning System
IMA	Integrated Modular Avionics
IP	Internet Protocol
IRIG	Inter-Range Instrumentation Group
Kpbs	Kilobit per Second
MAC	Medium Access Controller
Mpbs	Megabit per Second
NTP	Network Time Protocol
OMNET++	Objective Modular Network Test-bed in C++
P2P	Peer to Peer

PHY	Physical
PI	Proportional Integral
PPM	Parts per Million
PPS	Pulse per Second
PTP	Precision Time Protocol
PTPd	PTP daemon
RGMII	Reduced Gigabit Media Independent Interface
syncE	Synchronous Ethernet
SW	Switch
SWaP	Size Weight and Power
TSN	Time Sensitive Network
UTC	Universal Time Coordinated
WAN	Wide Area Network
WR	White Rabbit

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

FPGA Implementation of IEEE 1588 Precision Time Protocol for Avionics Systems

Muhammed Emin YABACIOĞLU

Department of Avionics Engineering
Master of Science Thesis

Supervisor: Prof. Dr. Şeref Naci ENGİN
Co-supervisor: Dr. İbrahim HÖKELEK

Technological developments have been allowed next-generation electronic devices to be more capable. A new architecture called Integrated Modular Avionics (IMA) has emerged because of the challenges to integrating with latest electronics into the conventional avionics architecture. Thanks to IMA, a flexible architecture has been revealed with a software and hardware pool that can perform various tasks. Besides the fact that each avionics performs its task, it is a more critical requirement that the data coming from them is interpreted together and presented to the user. It is necessary to ensure that the receiving time of data is the same as it is interpreted on a central computer. Therefore there is a need for a common time notion between avionics equipment. This requirement must be resolved within the existing IMA architecture without additional overhead. Taking advantage of the Ethernet-based ARINC-664 network used in IMA, it is believed that it would be beneficial to apply the IEEE 1588 time synchronization protocol, which is also Ethernet-based, to this architecture. In this thesis, where the IEEE 1588 protocol is implemented on Field Programmable Gate Array (FPGA) and its performance tested on a real network, a new feature called Rate Correction is proposed to improve the precision provided by the Precision Time Protocol (PTP).

The performance of the PTP design implemented on the FPGA is measured in various test scenarios with three different network topologies. As discussed in the Results section, it is seen that the Rate Correction feature contributed significantly to the PTP synchronization performance. The implemented PTP node synchronization

performance is measured using scenarios such as only PTP traffic or Ethernet background traffic in the network. According to the results, the Rate Correction feature will be improved to increase the synchronization performance in the following study.

Keywords: Avionics systems, Precision Time Protocol, time synchronization, fpga, rate correction feature

ÖZET

Aviyonik Sistemler için IEEE 1588 Hassas Zaman Protokolünün FPGA Uygulaması

Muhammed Emin YABACIOĞLU

Aviyonik Mühendisliği Anabilim Dalı
Yüksek Lisans Tezi

Danışman: Prof. Dr. Şeref Naci ENGİN
Eş-Danışman: Dr. İbrahim HÖKELEK

Teknolojik gelişmeler yeni nesil elektronik sistemlerin daha kabiliyetli olmasına imkan sağlamıştır. Bu elektronikler kullanılarak oluşturulan aviyoniklerin eski nesil aviyonik mimarilere entegrasyonu zor olduğu için Entegre Modüler Aviyonik (EMA) adında yeni bir mimari ortaya çıkmıştır. EMA sayesinde farklı görevleri yapabilecek bir uygulama ve donanım havuzu ile esnek bir aviyonik mimari ortaya koyulmuştur. Her aviyonik birimin kendi görevini yapmasından ziyade bu birimlerden gelen verilerin birlikte yorumlanması ve kullanıcıya sunulması daha önemli bir gereksinimdir. Verilerin merkezi bir bilgisayarda yorumlanırken aynı zamana ait olduğunun garanti edilmesi gereklidir. Bu durumda birlikte çalışması gereken aviyonik birimlerin ortak bir zaman bilgisine sahip olması gereksinimi ortaya çıkmıştır. Bu gereksinimin varolan EMA mimarisi içinde, ek yük getirmeden, çözülmesi önemlidir. EMA'da kullanılan Ethernet tabanlı ARINC 664 olarak adlandırılan ağın sunduğu avantaj kullanılarak, yine Ethernet tabanlı olan, IEEE 1588 zaman senkronizasyonu protokolünün bu mimariye uygulanmasının faydalı olacağı düşünülmüştür. IEEE 1588 protokolünün FPGA üzerinde gerçeklenmesi ve gerçek bir ağ üzerinde performans testi yapılan bu tez çalışmasında, Hassas Zaman Protokolü'nün (HZP) sağladığı hassasiyeti iyileştirmek için Oran Düzeltme adında yeni bir özellik önerilmiştir.

Aviyonik birimlerin zamanlarının senkron olmaması, çalışmaya başlama zamanlarının veya yerel zamanlarını oluştururken kullandıkları osilatörlerinin farklı olması olmak üzere iki temel sebebi vardır. Farklı zamanlarda çalışmaya başlayan birimlerin senkronizasyonu HZP'de tanımlanan Gecikme İstek-Cevap mekanizması

ile yapılabilmektedir. Zamanlar HZP ile eşitlense bile osilatörlerin farklı olmasından dolayı birimlerin arasında tekrar zaman farkı oluşacaktır. Bu çalışmada önerilen Oran Düzeltme özelliği bu problemin çözümü için geliştirilmiştir.

FPGA üzerinde gerçeklenen tasarımın üç farklı topolojide, farklı test senaryoları ile performans ölçümü yapılmıştır. Sonuçlar kısmında da anlatıldığı gibi Oran Düzeltme özelliğinin HZP performansına önemli ölçüde katkı sağladığı görülmüştür. Ağda başka Ethernet trafiği varken HZP performansını ölçmek amacıyla yapılan testler, senkronizasyon performansının düştüğünü göstermiştir. Alınan sonuçlara göre bir sonraki çalışmada trafik altında düşen senkronizasyon performansını arttırmak için Oran Düzeltme özelliğinde geliştirme yapılması hedeflenmektedir.

Anahtar Kelimeler: Aviyonik sistemler, Hassas Zaman Protokolü, zaman senkronizasyonu, fpga, oran düzeltme özelliği

1
INTRODUCTION

This section includes the motivation for studying PTP and our contributions on the design and implementation of PTP on FPGA and the organization of the thesis.

1.1 Literature Review

The word "avionics", which is derived by combining the words "aviation" and "electronics", is a term used to describe all electronic devices used in aircraft, such as flight control, flight management, navigation, and communications. Avionics devices that enable aircraft to perform their tasks safely and appropriately have been used since the early days of the aviation industry. The aircraft's capabilities have expanded to meet new and enhanced missions, more reliable, and fault-tolerant requirements through avionics. In addition, the mission scope of aircraft has greatly expanded [1].

Aircraft designed for various military or civilian missions have different capabilities depending on their intended use. In recent years, technology has begun to be developed to produce fifth-generation fighter aircraft such as the F-35, which have the ability to perform multiple roles in the aviation ecosystem. The F-35 is designed to leave the tactical tasks and decision making to the pilot. This principle has resulted in aircraft avionics components of the aircraft being designed to perform more missions. With the use of new generation avionics, a set of sensor data placed on the aircraft has provided the possibility for autonomous flight. Although pilots decide by interpreting the data they read from separate indicators or displays in previous fighter aircraft with their means, thanks to the advanced technology in the latest generation fighter aircraft, they only get involved in decision making [2].

While many of the controls in the earliest aircraft were performed with mechanic types of equipment, the radio used for communication is a simple example of the avionics systems in these aircraft. The architectures used in the previous aircraft, which were created by adding some basic features such as navigation, are not sufficient

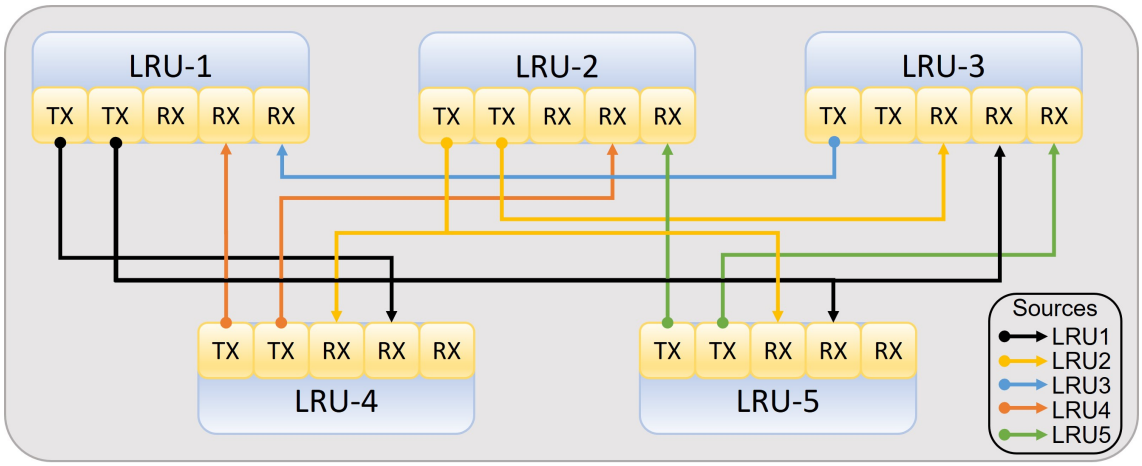

Figure 1.1 Example ARINC-429 data bus

to meet today's requirements. As in the example of the F-35, advanced software and hardware avionics are used in air-crafts to realize the versatile capabilities of the aircraft developed in recent years. In the next generation architectures consisting of mission computers and data buses, the main requirements such as high-speed data transmission and processing of this data, flexibility in reconfiguration, efficient use of power, and keeping weight at a minimum level have emerged [3]. In addition, recommended architectures for use in next-generation aircraft are expected to have features such as safety, redundancy in case of equipment failures, determinism, and real-time operation.

Until the 1990s, avionics systems are designed in accordance with federated architecture [4]. This architecture, which has been around from the beginning of avionics, consists of Line Replaceable Units (LRUs) connected over standard data communication buses such as ARINC 429 and MIL-STD 1553, where each LRU is dedicated to a specific mission. Since ARINC 429 has a "one mission = one computer" principle, the development and life cycle costs of developing and maintaining a large number of distinct LRUs are significantly higher. ARINC 429 data bus implements communication over serial lines.

Figure 1.1 shows an example ARINC 429 network. In this example, five LRUs represent different functions. The connection of LRUs for transmitting and receiving is made point-to-point with a twisted-shielded pair cable. A transmit port can send data to multiple receivers. Transmitting ports of safety-critical applications are separated due to the different certification levels [5].

ARINC 429 data bus consists of a transmitter and a receiver with up to 20 LRUs that support unidirectional communication by contrast with modern network protocols. The transmitter sends a message to the bus, and the receivers continuously monitor

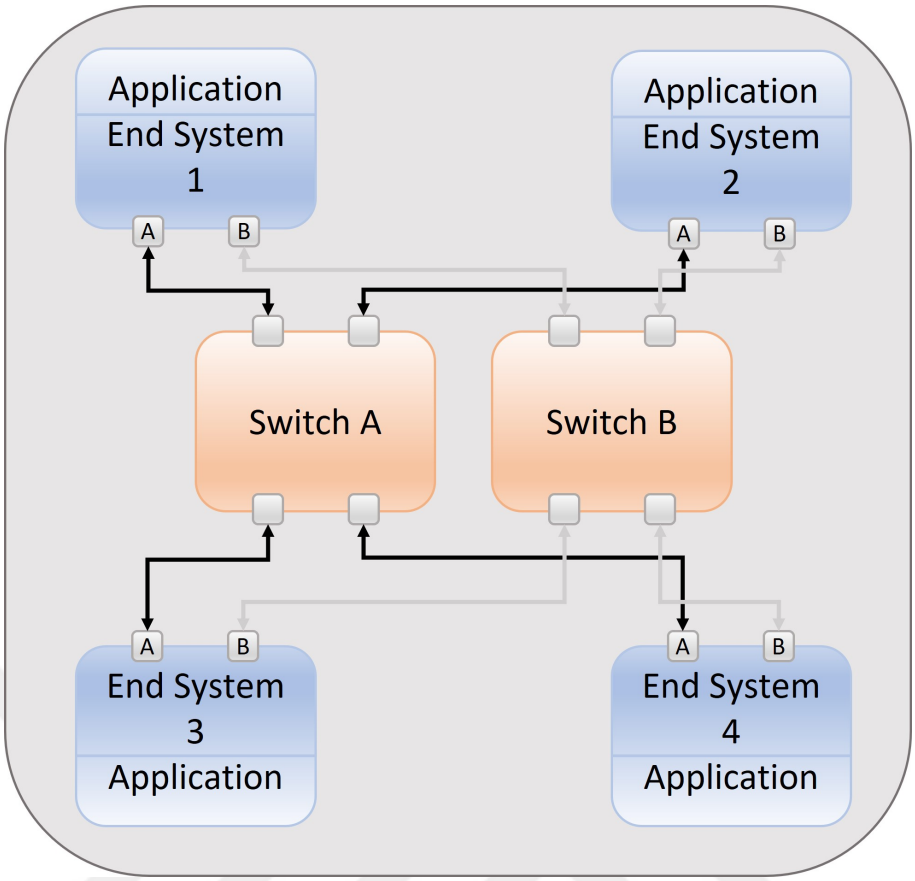

Figure 1.2 Example IMA architecture

the bus to detect messages and read the message that interests them. Since communication is unidirectional, each LRU must be connected to other nodes through its line to provide communication. The network becomes more complex with such busses, and more cabling is required. The messages consist of 32-bit data words. The data rate of the bus can be operated fast and slow at 12-24 kbps and 100 kbps respectively [6].

Improving technology has opened a new perspective in aerospace called Integrated Modular Avionics (IMA) architecture. Multiple avionics functions can run through different sources during operation, thanks to IMA architecture, which has a partitioned resource pool that includes such as shared computing, communications, and I/O interface [7]. ARINC-664 is a deterministic network protocol running over Ethernet and used in IMA architecture. Nevertheless, the ARINC 664 protocol inter-operates with the ARINC 653 protocol so that the system functions are time and space partitioned operating systems. ARINC 653 standard describes operating system and application program interface that produces safety-critical avionics systems. The IMA architecture, consisting of ARINC-664 and ARINC-653, ensures that avionics systems are simple, enables the implementation of software applications and hardware interfaces with the highest reliability requirements for avionics systems, and provides

complete flexibility to reconfigure system design. The IMA facilitates rapid prototyping using low-cost COTS components and can be easily upgraded beyond the initial implementation point [8]. An example IMA architecture that consists of ARINC 664 and ARINC 653 protocols is depicted in Figure 1.2.

In the part figured as application side on which the ARINC 653 protocol runs, the raw data from the sensors are transformed to Ethernet frames and transmitted to the End System (ES) side with PCI-e or another communication protocol. End systems are connected through ARINC 664 Switches (SW). The end systems are used to transforming raw data to Ethernet frames.The processed frames are transmitted to the network. ARINC 664 switch is the unit that provides the connection of the end systems communicating in the network. This unit performs routing of data traffic and runs a traffic policing mechanism so that the network operates deterministically. Thanks to ARINC 653, applications run by creating partitions can run on any end system, regardless of where they are installed. In this way, a modular structure is obtained. The Ethernet-based network, redundant topology, and guaranteed deterministic operation are some of the features that IMA architecture brings to avionics systems.

It is necessary to make arrangements in the architecture for adding a new feature to aircraft. Federated architectures supports low data transfer rates as it is designed for architectures using former technologies. Assigning a specific LRU for each function in federated architectures is one of the major drawbacks. Changes to aircraft avionics in federated architecture have negative consequences, such as architectural complexity, increased harness, maintenance, and operating costs. Because of the ARINC 664 protocol's Ethernet-based connection, the IMA architecture provides advantages such as modularity and reconfiguration. High data transfer and processing in avionics systems are became possible thanks to the IMA architecture that is evolved to solve these challenges.

Next-generation avionics systems designed with the perspective of IMA consist of dispersedly located units over an airplane such as sensors, actuators, controllers. The central computers, where the data read from the receivers on the aircraft are collected, are the data processing unit of the avionics systems in the aircraft. As with all distributed systems such as automation-based Industry 4.0 and Internet of Things (IoT) applications, time synchronization is a critical requirement for the avionics ecosystem [9]. The data fusion algorithm is conducted by the IMA's central computer to produce useful results from incoming data from avionics units located throughout the aircraft. Time synchronization between the central computer and the avionics components is necessary for the data fusion algorithms to operate properly [10].

The formation of a common time reference for IMA network components are possible with time synchronization of avionics applications. The network data captured by independent avionics units can be reconciled thanks to time synchronized avionics. Neglecting the importance of time synchronization creates conflicting data that is difficult for algorithms to process and merge [11].

Federated architecture generally provides time synchronization by using a hardwired connection between distributed LRUs and therefore requires a much higher wiring harness. On the other hand, an IMA architecture that has a high-speed data communication network facilitates the implementation of packet-based IEEE 1588 for accurate time synchronization. Lower Size, Weight and Power (SWaP) can be obtained by utilizing the PTP for time synchronization among avionics subsystems over a unified high-speed data communication network [12].

1.2 Objective of the Thesis

In this thesis, IEEE 1588 Precision Time Protocol, which is used to establish the time synchronization has been designed and implemented on FPGA. PTP synchronization and syntonization functions are performed by the implemented node. The end-to-end delay calculation mechanism proposed in PTP is used to calculate the offset between the nodes and to correct the time difference. A real-time Ethernet based test environment consisting of end systems and switches is used to illustrate the performance of PTP synchronization and the impact of the rate correction function.

1.3 Hypothesis

The implemented FPGA design allows time synchronization between the network nodes. The syntonization process is implemented for better precision that named rate correction in this thesis. This feature is not explained clearly in IEEE 1588 standard, so an implementation method has been practiced and tested its effect on synchronization precision. Also, performance analysis is performed with various network traffic scenarios. Different test scenarios are prepared depending on whether the Ethernet background traffic and rate correction feature is enabled or disabled. The average and maximum values of deviation in time synchronization between nodes are reported. The results show that time synchronization can be established at the nanosecond level using the end-to-end delay calculation mechanism. If the rate correction feature is disabled, time synchronization is not maintained. Then, when the utilization rate of network bandwidth increases, synchronization accuracy increases to the level of microseconds, resulting in asymmetric path delay.

1.4 Organisation of the Thesis

This thesis consists of five chapters. Chapter 1 gives general information about time synchronization requirements in avionics architecture. Chapter 2 contains related works and currently used time synchronization protocols for avionics. Chapter 3 introduces the IEEE 1588 standard synchronization mechanism, followed by an explanation of the system model and rate correction, a method for enhancing synchronization precision. Chapter 4 presents the FPGA design of PTP and numerical results for different test scenarios. Chapter 5 contains the thesis results and discussions, and presentations for future work.

2
TIME SYNCHRONIZATION OVERVIEW

In this section, time synchronization protocols are compared and related works on time synchronization with the PTP standard are explained.

2.1 Time Synchronization Protocols

Time synchronization is the process by which sub-components in the same network share a common notion of time. It is a fundamental requirement for aircraft avionics systems, as it is for other distributed systems. Different protocols with varying levels of accuracy are used to synchronize the time of units in distributed systems.

The time of distributed network nodes might differ in two ways: the origin of time and the frequency of the oscillator. Each node can be run at different times, so this causes asynchronous time of day information between nodes. Even if the network node's time of day value is the same at the beginning of the run, each node uses its oscillator for advancing its time information. So there will be an offset between local time information during operation due to the different operating characteristics of the oscillators [13].

In general, time synchronization methods are classified in two ways, hardwired or packet-based. Hardwired time synchronization uses point-to-point electrical signals that do not require the exchange of timing information between nodes. Packet-based time synchronization protocols are executed over a network to communicate between nodes. Synchronization between nodes is established by exchanging timing information with packets sent over the working network [14].

Architectures in which synchronization is provided in a hardwired solution have drawbacks such as lack of flexibility, increased weight due to cabling, and the impact of physical conditions on synchronization precision. On the other hand, such problems have been overcome in packet-based synchronization applications. However, two types of difficulties in such a synchronization protocol directly affect the precision

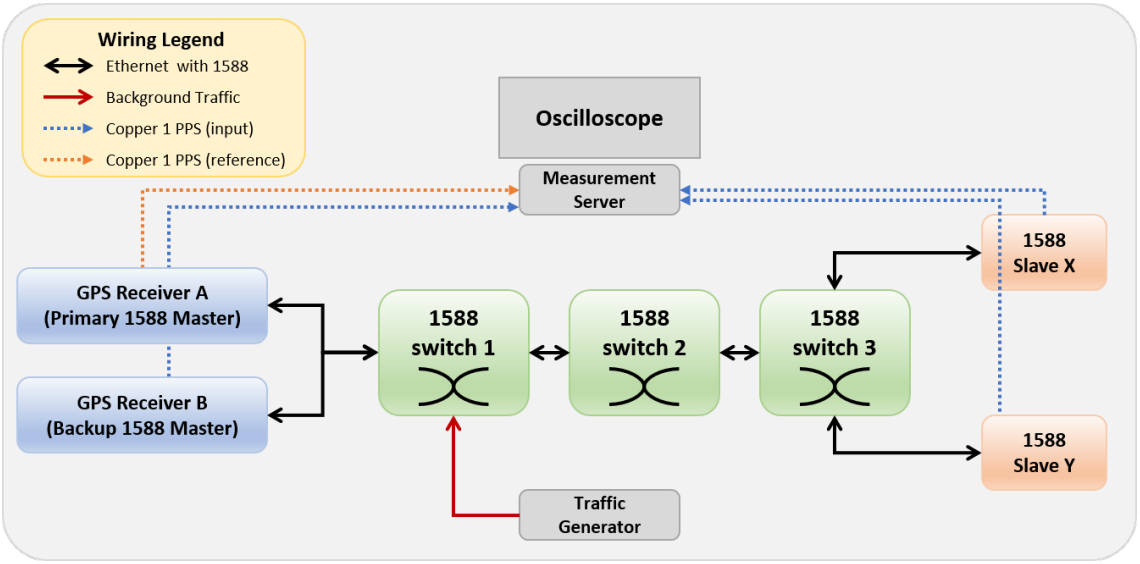

Figure 2.1 Example GPS and PTP architecture [17]

and accuracy. The first difficulty is the accurate timestamping process of the protocol packets, which are carried egress time from the transmitting node and the ingress time to the receiving node. The correct processing of the time information into the package is insufficient for an excellent synchronization result. The second challenge is to accurately measure the path delay that packets carrying time information for synchronization support as they are transmitted between nodes in the network [15]. Inter-Range Instrumentation Group (IRIG-B), Network Time Protocol (NTP) and Precision Time Protocol (PTP) are three well-known synchronization protocols [16].

The Global Positioning System (GPS) is used for time synchronization and positioning for general use. Any avionics units equipped with a GPS antenna can be synchronized with the satellite clock. Attaching the GPS receiver, which is a must for synchronization with the GPS, to each piece of equipment on the aircraft introduces additional difficulties in terms of weight and cost. In addition, the fact that the signals from GPS are easily affected by environmental conditions and are susceptible to interference is a disadvantage for time synchronization between avionics devices. Therefore, GPS is more suitable for inter-aircraft synchronization than for in-flight avionics synchronization. In today's avionics systems, utilizing GPS as the primary source of time information and an auxiliary protocol such as IRIG-B or PTP to establish synchronization between avionics devices in the aircraft is preferable [17].

Figure 2.1 shows an example architecture of time synchronization used combination of GPS and PTP. PTP is used for transferring time information between nodes. In [17] results, it is claimed that synchronization using the GPS signal increases synchronization precision, but transient behavior of GPS receivers in case of failure

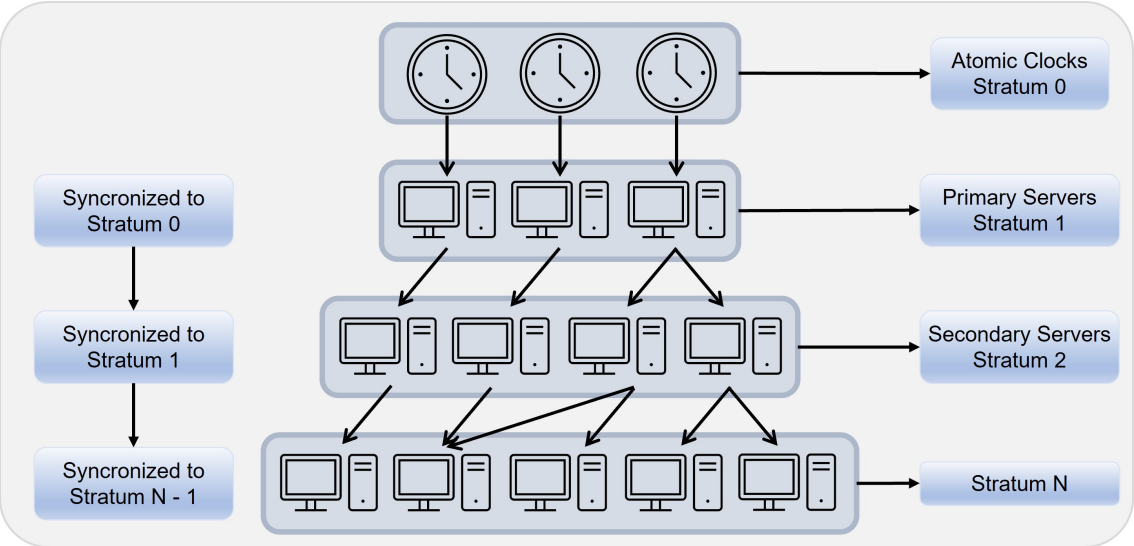

Figure 2.2 Example NTP architecture

needs to be thoroughly tested.

The IRIG-B protocol has been used in avionics systems to establish time synchronization via the hardwired method [18]. Time information between nodes is transmitted serially, either modulated or unmodulated. The modulated time information is transmitted as a sine wave, similar to Amplitude Modulation (AM) radio frequency, and millisecond synchronization accuracy can be achieved. The Direct Current (DC) level-shifted signal is used in the unmodulated transmission of timing information. It has been observed that synchronization can be achieved at nanosecond levels. The significant advantage of the modulated signal is that it can be transmitted over long distances without being affected by the transmission medium and without the need for extra reinforcements [19]. The IRIG-B protocol makes the architecture inside the aircraft complex and adds weight because a separate bus is used [16]. Also, since the time information is carried hardwired, the cables must be protected against electromagnetic interference (EMI) inside the aircraft.

NTP is one of the prominent protocols in packet-based time synchronization applications. The goal of NTP is to synchronize the times of computers in a Wide Area Network (WAN) such as the Internet or intranet [20]. It is a synchronization protocol based on the Internet Protocol (IP), implemented in software, and Ethernet is used for communication. The protocol, which works using the client-server model, processes time information at the application layer. Since the timestamping process of the protocol packets is done at the application layer, it is not possible to precisely timestamp the time information when they are transmitted to the physical line. Moreover, path delay is measured incorrectly due to the changes in the packets transmitting paths and network congestion. Such effects reduce synchronization

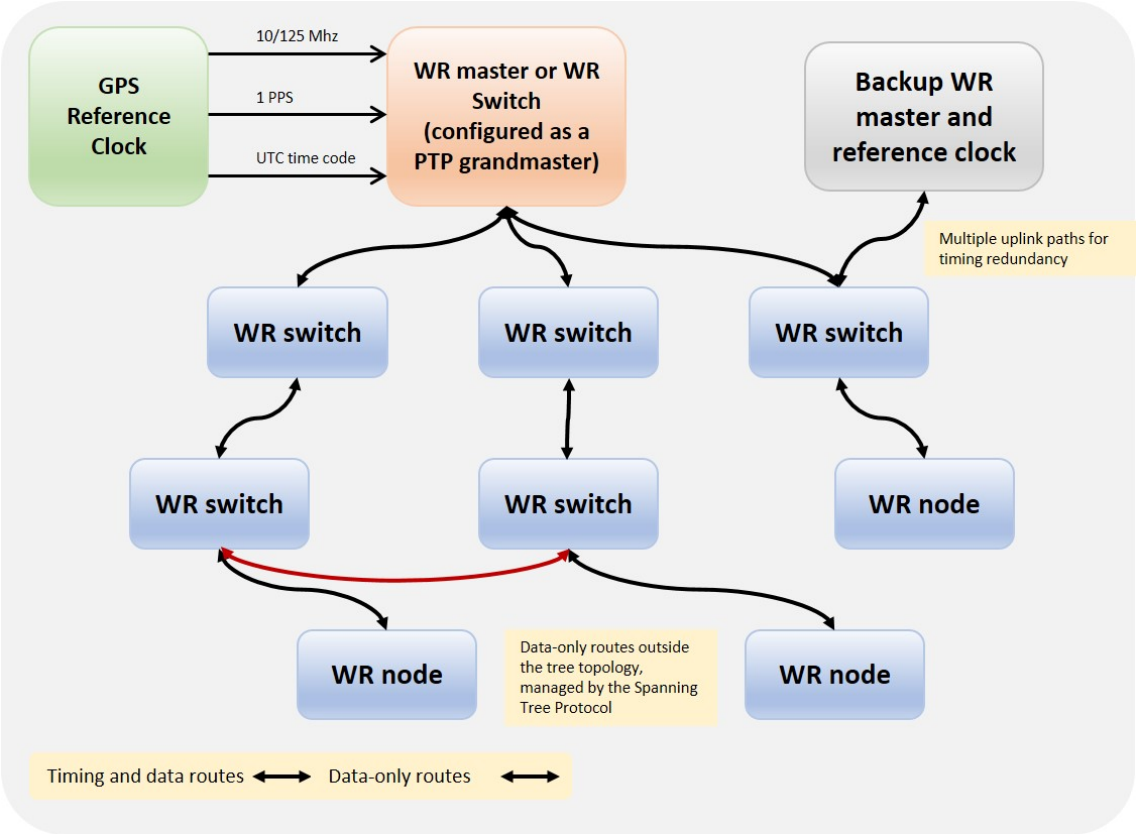

Figure 2.3 Example WR network [24]

precision and accuracy [21, 22]. Therefore, time synchronization with NTP is possible at millisecond levels. NTP is not preferred in aviation because avionics used in aircraft require more precise time synchronization [16].

As seen in Figure 2.2, the first time source is named Stratum 0, in subsequent units Stratum 2 or 3. Each level, called the stratum, is responsible for synchronizing a sub-unit. System time information is taken from the primary source, the atomic clock, and transmitted to the next servers in a hierarchical order.

Synchronization protocols that require a separate bus, such as IRIG-B, are not recommended in the IMA approach, which is utilized in the design of new generation aircraft's avionics architecture. Such a design would diminish the re-configurable feature of the IMA. While Ethernet-based operation of NTP provides flexibility, its low synchronization precision and accuracy is a disadvantage for use in avionics. IEEE 1588 - PTP protocol is intended to meet the requirements of the IMA architecture, such as high precision time synchronization, flexible architecture design [23].

IEEE 1588, also known as PTP, achieves sub-microseconds precision by using similar time exchanging method with the NTP. However, there are several fundamental differences between NTP and PTP. The main difference is that PTP supports hardware

implementation as well as software or can be operated with hardware support. The hardware feature allows timestamping process as close to the physical layer as possible. This feature of PTP recovers the timestamping precision from jitter and processing delays. In addition, PTP is defined for synchronizing nodes on local area networks (LAN), as opposed to NTP, which is used for synchronization with Internet clocks. This work is based on version 2 that published in 2008 [25].

White Rabbit (WR), which is based on PTP and is a project of the European Organization for Nuclear Research (CERN), is a version of PTP with increased synchronization precision. In this project, synchronization can be provided with PTP operating together with the Synchronous Ethernet (SyncE) protocol with sub-nanosecond level [24]. Since the use of a separate protocol such as SyncE and the precision provided by PTP are sufficient, there is no need to use the WR project for avionics architectures. Figure 2.3 shows the example of a WR project network.

The protocols described above, such as IRIG-B, NTP, and GPS, are unsuitable for the IMA architecture in next-generation aircraft. The flexibility provided by the IEEE 1588 will bring many advantages in avionics systems, especially in IMA architecture. Therefore, in this study, we have focused on the implementation and testing of PTP to represent the operation of avionics systems.

2.2 Related Works

There are various studies in the literature on time synchronization with PTP. However, these studies generally focused on comparing PTP with other time synchronization protocols, explaining the working method of synchronization flow, or researching the synchronization precision and accuracy achieved depending on the applied environment such as software or hardware. In the performance tests, only the synchronization result is considered, and the negative or positive effects that changed the precision of the synchronization are ignored. For example, a few works focus on performing tests to synchronization precision when there is other Ethernet traffic in the network where the protocol is running. Calculating the offset difference between nodes and making corrections have been prioritized. Although rate error is one of the significant sources of synchronization errors, it is occasionally overlooked in distributed clock synchronization algorithms [26].

In [27], a study in which time synchronization with PTP is implemented and tested in a software-based manner, software based PTP daemon (PTPd) is run. This study examines the synchronization precision that can be achieved using the IEEE 1588 protocol without special hardware support. The tests were carried out on the network

Table 2.1 Test results [28]

Network Setup		Time Offset [us]			Delay Spikes [us]	
		Mean	Mode	Std Dev.	Lower	Higher
Cross Cable		36.6	38	1.67	30	39
100 Mb Switch	PTP only	50.5	51	1.61	44	54
	CPU-jam	55.6	55	2.39	44	62
	FTP	50.8	60	5.72	96	131
10 Mb Switch		168.2	169	2.49	157	172

where the VXI Technology EX1048 device, which is embedded Linux operating system and running PTP as hardware-assisted, is running as a slave node to the test device Agilent LXI IEEE-1588 Demonstration Kit. With the hardware-assisted clock running on the FPGA, it is possible to obtain timestamps with microsecond precision, allowing frequency correction to be used instead of the Linux software clock. During tests, it has been observed that the PTPd is synchronized to the master clock with an accuracy of 10 microseconds(µs) precision.

In [28], where PTP is implemented as software, the conditions that affect the precision of synchronization are studied. Two computers with a Linux operating system were used to exchange PTP packets for the tests. One computer with a Windows operating system was used for the network traffic between the computers. Four situations were observed during the tests performed. These are the synchronization time of the devices that start working with a certain time deviation; the effective synchronization value that can be achieved; events and anomalies that require detailed analysis; and critical situations that can affect the synchronization. First, when two computers are directly connected, and there is no other traffic in the network, the three phases until synchronization is achieved were defined: the start phase, the non-linear phase, and the linear phase. The accuracy of synchronization was measured around 37±3 µs. Second, two computers are connected through a 100 Mpbs switch. An offset of 51 µs was measured without any effect, such as other traffic in the network and no other operation on the Central Processing Unit (CPU). Then, in this topology, the CPU-jam software that keeps the CPU busy was run on PC -1 and PC -2, so that very few spikes were seen in the synchronization results. By connecting PC -3 to this topology, %60 of the bandwidth was used, and it was observed that synchronization was achieved around 60 µs. In another test, only disk-jam software was run, which forces the PC to write data on its hard disk continuously. In this scenario, the situation of jitter in timestamping process on the CPU was applied, and synchronization was calculated ±50 µs. In the last scenario, where the Ethernet switch was operated with 10 Mpbs and %30 bandwidth was used, 169 µs of synchronization was achieved. The results for all test cases can be found in Table- 2.1.

In [29], PTP is implemented using FPGA, DM9000 and DP83640. This design is a combination of hardware and software to implement IEEE 1588 protocol. The FPGA used is Altera's Cyclone 2 product, which has a softcore processor embedded in it. The DM9000 is a chip with an Ethernet MAC controller, while the DP83640 is a chip manufactured by National Semiconductor specifically for the IEEE 1588 protocol. The PTP was run with the DP83640 chip controlled by the FPGA. Synchronization was ensured by adding a timestamp to the protocol packets used when traversing the physical layer (PHY). The test topology consists of nodes containing two DP83640 chips directly connected to each other. The 1 PPS signal from the nodes was compared using an oscilloscope to measure the synchronization precision. The tests were repeated by changing the synchronization frequency to 1, 2, and 4 seconds, respectively. According to the results, it is possible to perform time synchronization with sub-microseconds level by using the PTP in this manner.

It has been observed that in [28–30] papers are limited to comparison timestamping methods for synchronization. Asymmetric path delay effect is not considered which is occurred as a consequence of network traffic rate on synchronization level.

Accurate measurement of time difference is directly related with nodes running frequency [26]. PTP only specifies a way to estimate the offset of a slave clock frequency in reference to a master clock frequency. The implementation of clock servo controller is not explained in the standard. This improvement to better precision is left to the discretion of the designer [31]. In [31], simulations based on $OMNeT++$ focus on oscillator noise simulation and oscillator synchronization using PTP. The simulation results show that that "Clock Servo" module of PTP node has increased the precision of synchronization. This module of PTP node model runs proportional-integral (PI) algorithm, which is used to calculate oscillating differences. The [32] paper adopts to create a model of clock drift rate and uses this drift rate to adjust time and analyze stability. Results show that oscillators drift rate calculating reduces the effect of network delay and jitter and achieves high precise and accurate time synchronization.

In [33] syntonization and synchronization functions of PTP completely realized in FPGA. The syntonization term is used for frequency synchronization. This feature runs a PI servo controller for synchronization. The aim of [33] such an implementation is to provide high accuracy and a robust system clock that can be driven by a simple oscillator.

3
SYSTEM MODEL of PTP with RATE CORRECTION

This section contains information about the IEEE 1588 standard. First, an example PTP network for synchronization is discussed. The types and characteristics of the clocks involved in synchronization in the PTP network are explained. Then, the method of PTP synchronization is explained. Finally, information is given about the Rate Correction feature that we recommend to improve the synchronization precision provided by the standard.

IEEE 1588 is a packet-based synchronization protocol used to synchronize the time of different devices on the same network. System integration is easier and less costly than other synchronization protocols such as IRIG-B, NTP, GPS [9]. PTP is used in many different areas such as test and measurement systems, automation systems, telecommunications, global positioning systems. Nanosecond level time synchronization can be established by implementing the standard [34].

3.1 IEEE 1588 Domain and Clock Types Overview

The network where the PTP protocol runs is called the PTP domain. In PTP domain, time synchronization occurs between master and slave nodes. The node working as the master is the source of time information. Nodes that are synchronized with the master node are called slave nodes. PTP Nodes working as master or slave have been separated into three types: Ordinary, Boundary, and Transparent Clock. Ordinary Clock has a single port, such as end- system in any network [35]. This port can operate in master or slave state. Boundary Clock is the name for the PTP node that has two or more ports. While one of its ports operates in slave state, the others switch to master state. It is similar to a network switch that connects the end systems in a network. The Transparent Clock, on the other hand, has multiple ports and is similar to the Boundary Clock but is not synchronized with the master node. It adds packet processing delay to forwarded PTP packets. The ports of the Transparent Clock do not change to a different state.

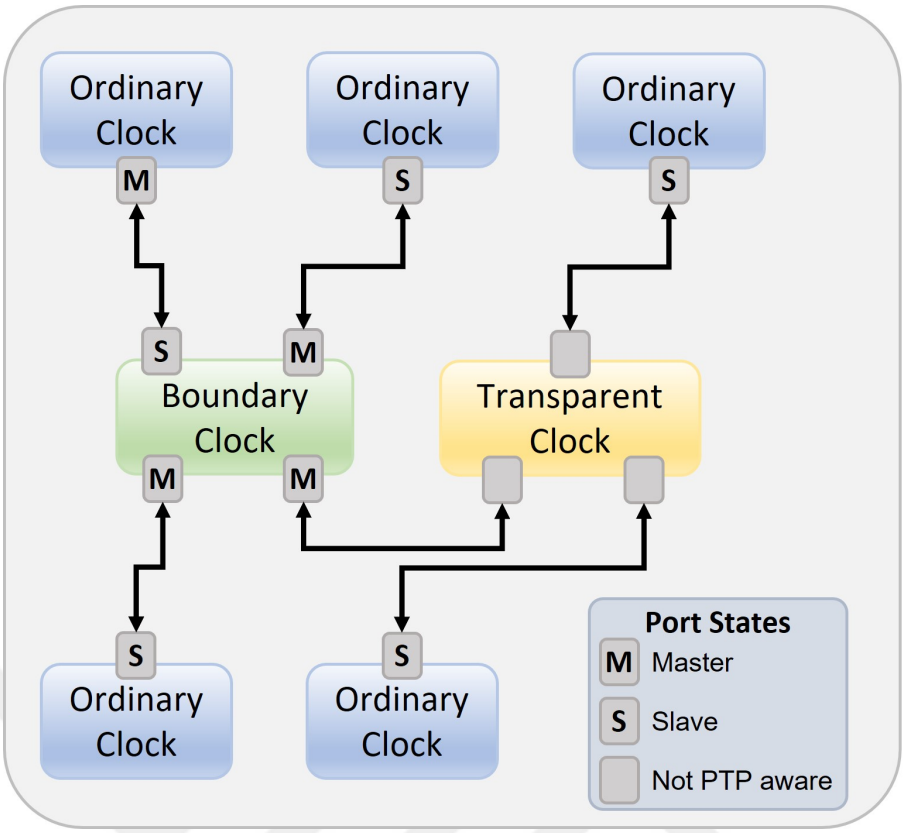

Figure 3.1 Example types of PTP clock network

Figure 3.1 depicts an example of a PTP network configured using different PTP clock types. In this figure, ordinary clock whose port is set as master can refer to an external source such as GPS and is assumed to have the most precise and accurate clock in the PTP domain. While the boundary clock's port works as a slave connected to the master, the other ports become the master and synchronize other connected nodes. The PTP domain in this example demonstrates the point-to-point (P2P) synchronization method. Thus, there is a sole master-slave relationship between the two connected nodes. In the PTP domain where synchronization is done as end-to-end (E2E), there is one master node, except for redundancy and reliability improvements, and all other nodes are synchronized directly with a master. The comparison of P2P and E2E is explained in the following section. In this thesis, E2E method has been used.

The IEEE 1588 protocol performs two main processes to achieve synchronization. First, the master-slave hierarchy is established between the nodes. Then, the packet exchange protocol begins for time synchronization between nodes.

The process in sub-clause 9, defined in the standard as the Best Master Clock Algorithm (BMCA), establishes the master and slave relationship between nodes in the PTP domain. During the execution of BMC Algorithm the parameters such as the precise and accurate operation or the time source reference related to the quality of the clock

resources are compared. As a result of this comparison, the node with the most accurate and stable clock moves to the master state. Other nodes in the PTP domain become slaves. If the master node fails during the execution of the synchronization in the PTP domain, the system can independently select a new master, thanks to the BMCA. This ensures that time synchronization is maintained without degradation, even if the master fails. The Alternate Master (subsection 17.3) and Grandmaster Clusters (subsection 17.4) features have been defined to reduce the time it takes for all nodes in the network to be re-compared to the BMCA and for a new master-slave hierarchy is created [36]. In the Alternate Master extension, the candidates to become masters are defined as alternate masters, and during the execution of the BMCA, a new master is selected by comparing only these alternates instead of all nodes. In the Grandmaster Cluster option, each slave node exchanges packets for synchronization with all possible master nodes but synchronizes only with the time information it receives from the real master. If the real master fails, the information from the next best candidate master is used, and synchronization continues without interruption.

It is not acceptable for avionics systems to perform this dynamic decision-making process during operation. For this reason, instead of BMCA, the node to be master should be defined statically, and the architecture should be designed redundantly if necessary. Since this thesis is about the time synchronization of avionics systems, the master and slave features of the nodes are defined statically without using the BMCA.

3.2 Synchronization Mechanism

The time synchronization starts after the master-slave hierarchy is established in the PTP domain. In order to achieve time synchronization, it is necessary to calculate the time difference between the nodes and the path delay of the packets. There are two options, Peer Delay Mechanism (P2P) and Delay Request-Response Mechanism (E2E), to calculate these values by the slave node. While the P2P method calculates path delay between each node in the PTP domain peer to peer, the E2E mechanism calculates path delay end-to-end. When the P2P mechanism is used, synchronization results may be more accurate. If this mechanism is utilized, all nodes in the network must be PTP-aware, which increases the cost and makes the implementation more complex. In this thesis, the E2E mechanism has been used because it is more straightforward in terms of implementation.

Synchronization with the Delay Request-Response mechanism uses four message types. Sync, Follow-Up, Delay Request, and Delay Response. The Sync and Follow-Up messages are used to calculate the time difference between slave and master nodes.

The Delay Request and Delay Response messages are used to calculate the path delay of packets transmitted between nodes. The Follow Up message is optionally used for cases where it is not possible to process the timestamp during the transmission of the Sync message. The situation where two messages are used is called a two-step operation, and only a Sync message is sufficient is called a one-step operation. In the implementation of PTP by the hardware method, it is possible to transmit packets by adding time information while transmitting to the physical layer [37]. Since this study includes the implementation of PTP entirely on FPGA, the one-step operation is used so that Follow-Up message is not required.

Sync and Delay Response messages are sent by the master, and Delay Request messages are sent by the slave. The flow of the synchronization process in one step is shown in Figure 3.2. The protocol applied to synchronize the local time between two nodes is as follows.

- Sync message is sent from the master node to the slave node and the egress time of packet from the master node as t_1 is written into the packet.
- The Slave node records the ingress time as t_2 when the Sync message is received.
- After Sync message is received by the slave node sends the Delay Request message to the master node and records sending time as t_3.
- When the master node receives Delay Request message, it sends Delay Request ingress time to slave node in the t_4 Delay Response message.

Where Δ_{MS} shows path delay from the master node to the slave node, Δ_{SM} shows path delay of the opposite direction.

The slave node calculates the time difference and the path delay, using the recorded time information in Equation 3.1 and Equation 3.2, respectively.

$$\Delta t = t_2 - t_1 \tag{3.1}$$

$$\Delta_{MS} = \frac{(t_4 - t_1) - (t_3 - t_2)}{2} \tag{3.2}$$

The equation 3.2 is calculate properly path delay value between two nodes if the path delay is symmetrical in both directions.

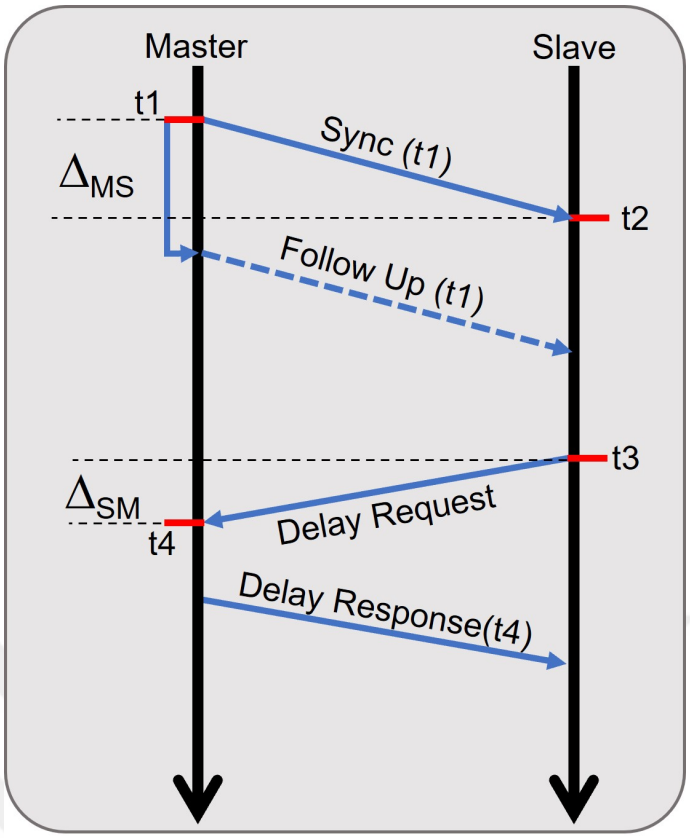

Figure 3.2 Synchronization mechanism message sequence

Since it is not possible to change the time information in the synchronization packet transmitted to the physical layer during transmission, it must be transmitted to the slave node as soon as possible. When the time difference is corrected with the time information in the packet sent by the master node, the time difference remains as a path delay in between as the master node's time continues to advance until the packet arrives. Synchronization process is completed by calculating the path delay with the Delay Request-Response messaging, which is performed after the Sync packet is received. However, these packets are sent in opposite directions and represent the path delay values in different directions. There is no guarantee that the Delay Request packet sent in the opposite direction is exposed to the same delay to calculate the path delay suffered by the Sync packet sent from the master to the slave. This situation, called delay asymmetry, significantly affects the accuracy of synchronization. In this thesis, tests are performed on the effects of asymmetric path delay, which are presented in Section 4.2.

The accuracy and precision of the time information written on the packets are as important as the correct calculation of the time differences. Figure 3.3 shows when timestamps can be added. The transmission times of packets transmitted from the application to the physical layer vary depending on the running application. In this

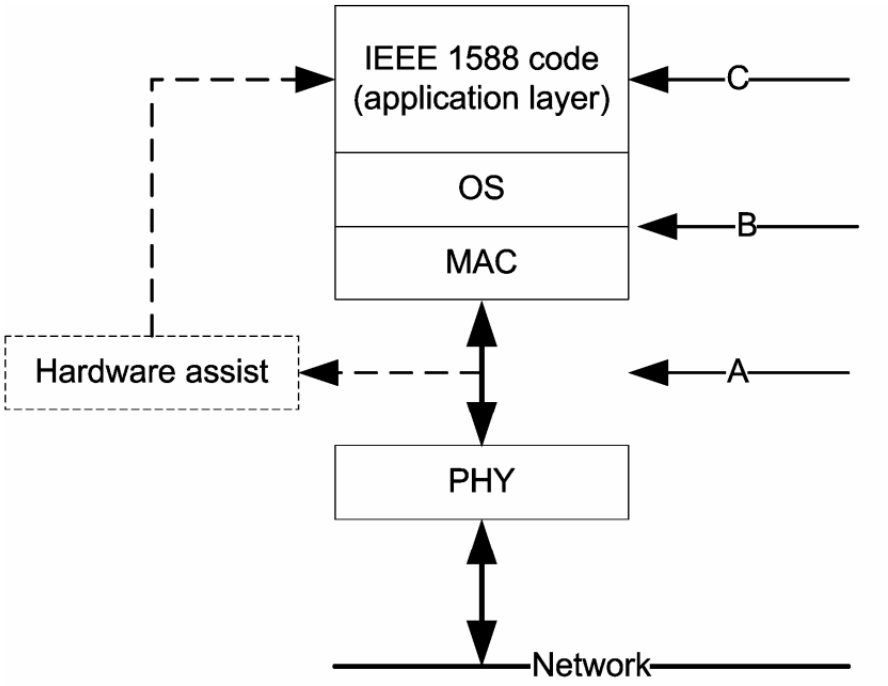

Figure 3.3 Possible timestamping points [25]

case, this will cause variations in the path delay calculation. As the stamping location approaches the physical layer, the accuracy increases as the variability in transmission time decreases. In this thesis, PTP is implemented on FPGA so that the timestamp can be inserted to the packet at point B, as shown in Figure 3.3. Since the packets go through the same operations in the MAC and PHY layers after this point, the delays do not change dramatically.

3.3 Rate Correction for Improving Synchronization Precision

PTP synchronization protocol described earlier solves the time difference that occurred as a result of the different starting points at which the nodes begin their time. However, even if the nodes time is equal, each node continues to increase its time information by using its oscillator after synchronization. The time difference is increased until the following synchronization sequence.

Oscillators generate the reference clock signal for the operation of electronic devices. The stability of oscillators varies depending on the type of production material and the environmental conditions. The variation caused by these effects is expressed as Parts Per Million (PPM). Consider the case where the frequency of the oscillators used by the master and slave nodes is 100 MHz, and the deviation is +-50 PPM. Assuming that the master node is at 100 MHz without errors, the oscillator generates 1 million clock pulses of 10 ns each in 1 second. Assuming that the slave oscillator is running at the

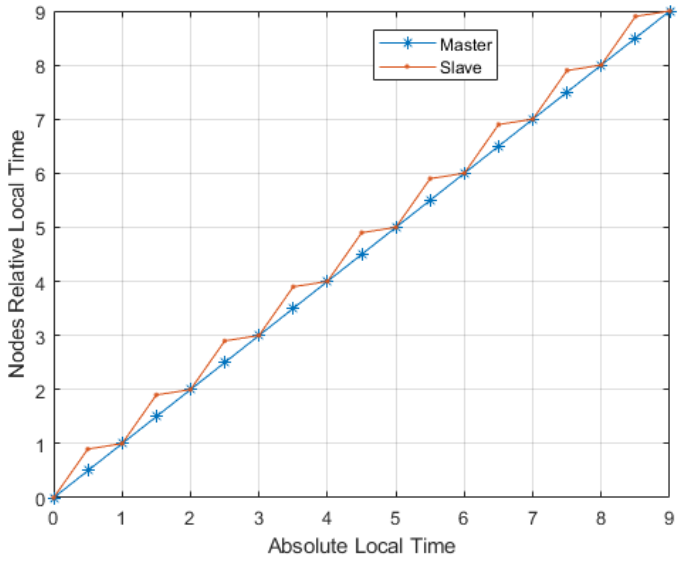

Figure 3.4 Oscillator PPM effect on synchronization

same time with an error of +50 PPM, it will generate 1005000 pulses by overclocking 5000 pulses, according to Equation 3.3. $CP_{variation}$ is the variation of clock pulse, $f_{operating}$ is the operating frequency of the node, and PPM is the instability of the oscillator. As a result, the time information of the slave node is generated as 1 second and 50 µs by using Equation 3.4. Δ_T indicates the time deviation of node, and $T_{operating}$ indicates the operating period of the node.

$$CP_{variation} = \left(\frac{f_{operating} * PPM}{10^6} \right) \tag{3.3}$$

$$\Delta_T = CP_{variation} * T_{operating} \tag{3.4}$$

Nodes have the same time information after synchronization, whereas the difference reaches 50 µs until the next synchronization interval of one second. Figure 3.4 is a graph showing the result of the oscillator error effect between synchronization intervals. It is necessary to detect and correct the different operations in the oscillators in order to minimize this effect.

Apart from the calculation of the time difference, a process called syntonization is defined in IEEE 1588 to ensure that the times of the nodes increase in a similar way, in sub-clause 12. Syntonization, which means "tuning," is recommended for synchronizing the slave node's clock frequency with the master node's frequency. This process is not emphasized enough in the standard. The equation to calculate this ratio

recommended by the standard is given in equation 3.5.

$$N = \frac{< t_{1,N} - t_{1,N-1} >}{< t_{2,N} - t_{2,N-1} >} \tag{3.5}$$

Where $t_{1,N}$ and $t_{2,N}$ represent the time information values of the last received sync message, respectively. The $t_{1,N-1}$ and $t_{2,N}$ values show the values related to the previous received sync message. After this process definition, it is stated that in the standard such syntonization mechanisms are out of scope.

The Rate Correction feature is defined in this thesis and proposed as an alternative to the syntonization process in the standard. The effect of this feature on synchronization precision is explained in the results section.

Similar to the process in the standard synthonization feature which is run by reference to Sync messages, works with a counter running on the slave node. In this function, which is operated by manipulating the counter values without changing the clock frequency settings of the nodes, there is no interference with the operation of the oscillators. The Sync Interval, T_{SI}, value currently available in the slave node specifies the interval of Sync packets sent by the master. The ratio of T_{SI} and f_{clk} represents the number of clock pulses, α_{SI}, corresponding to the Sync interval value. Equation 3.6 illustrates this calculation.

$$\alpha_{SI} = \frac{T_{SI}}{f_{clk}} \tag{3.6}$$

When the slave node receives the sync packet, it resets the counter and saves the current value. The variable β_{SI} in Equation 3.7 shows the counter value recorded by the slave node. The difference between α_{SI} and β_{SI} gives information about the operation of the slave node's oscillator relative to the master node. The pulse difference between the nodes is expressed as Δ_{CP}. Since there is no reference for the first Sync packet on the slave node, the value Δ_{CP} that starts calculating after the second Sync packet is calculated eight times to reduce the possibility of error. According to the Equation 3.7, it means that the master node's oscillator oscillates faster, when the α_{SI} value is higher, and when the β_{SI} value is higher, the slave node's oscillator oscillates faster. The value found by calculating the arithmetic mean of Δ_{CP} is expressed as $\overline{\Delta_{CP}}$. This value shows the clock deviation rate for the slave node. The

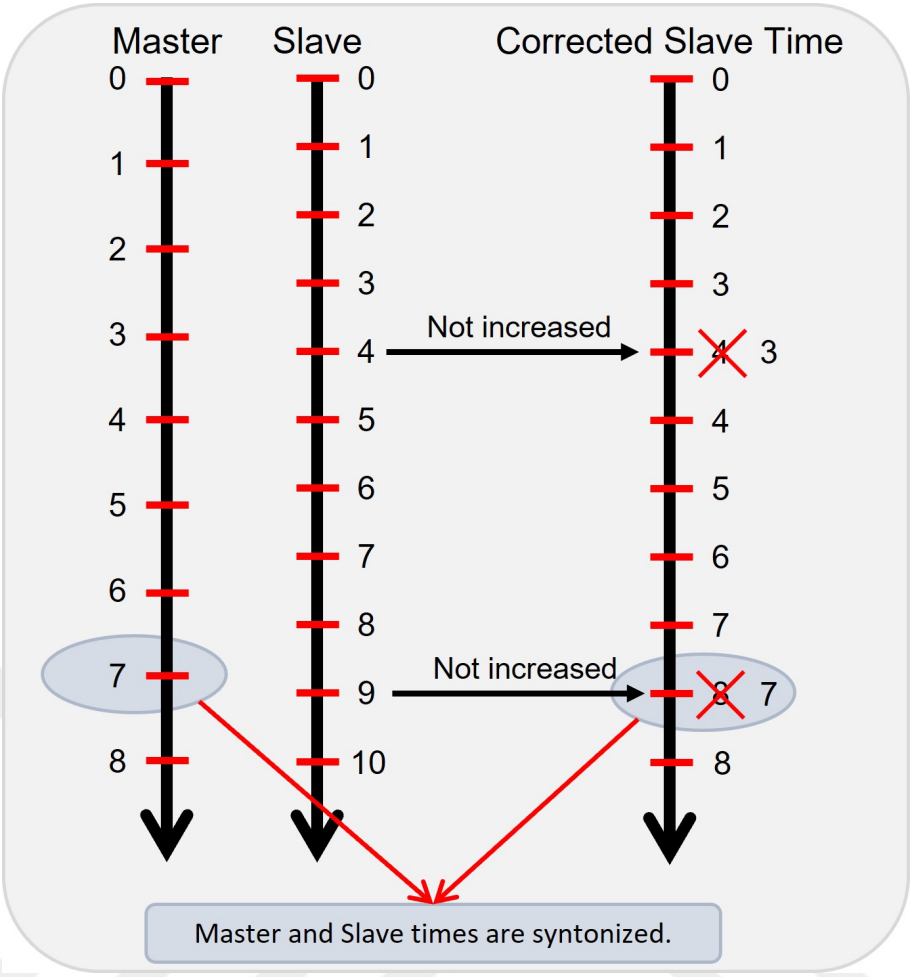

Figure 3.5 Example rate correction process

relevant calculations provided in Equation 3.8.

$$\Delta_{CP} = \begin{cases} \alpha_{SI} - \beta_{CP}, & \alpha_{SI} \geq \beta_{CP} \\ \beta_{SI} - \alpha_{SI}, & \alpha_{SI} < \beta_{CP} \end{cases} \tag{3.7}$$

$$\overline{\Delta_{CP}} = \frac{\Delta_{CP,1} + \Delta_{CP,2} + \Delta_{CP,3} + \ldots + \Delta_{CP,N}}{N} \tag{3.8}$$

Equation 3.9 is used to calculate the correction interval, γ_{CI}, which is required to correct the counter value of the slave node. When the slave node's oscillator is faster, the time information counter will not be incremented at that clock after γ_{CI} clock pulses in this node. When the oscillator of the slave node is slower, the time information counter will be incremented twice in that clock after γ_{CI} clock pulses in this node. Thus, after T_{SI} time, the times of the two nodes will be increased by an equal amount.

$$\gamma_{CI} = \frac{\alpha_{SI}}{\Delta_{CP}} \tag{3.9}$$

Figure 3.5 shows an example of frequency difference measurement and correction of the slave node's node. The effect of the Rate Correction feature on synchronization is explained in the numerical results section.

4
FPGA IMPLEMENTATION AND NUMERICAL RESULTS

In this section, we provide information about the FPGA design of the IEEE 1588 PTP protocol and present the performance results using the FPGA implementation on the development boards. The various experiment scenarios are prepared by configuring several parameters such as synchronization interval, Ethernet background traffic, and turning the PTP rate correction feature on and off.

4.1 FPGA Design

Figure 4.1 shows the PTP block design consisting of 6 modules: Message Generator, Message Parser, Time Generator, Offset Calculator, Rate Correction, Port Timestamper, Mac Core. All development boards that run tests are used the same implemented design. Whether the nodes are master or slave is determined by the configuration.

In this study, the IEEE 1588 protocol is implemented on the NETFPGA 1G CML development board that includes an FPGA in the Kintex 7 family. Vivado is used as the synthesizer tool. The values related to the resource usage of the implemented design are reported as shown in Table 4.1.

Table 4.1 FPGA source usage of implemented design

Logical Unit	Lut as Logic and FF	
	#	%
Message Generator	5143	2.52
Message Parser	592	0.29
Time Generator	727	0.35
Rate Correction	352	0.17
Offset Calculator	2299	1.12
Port Timestamper	1630	0.79
Mac Core	909	0.44
Summary	11652	5.71

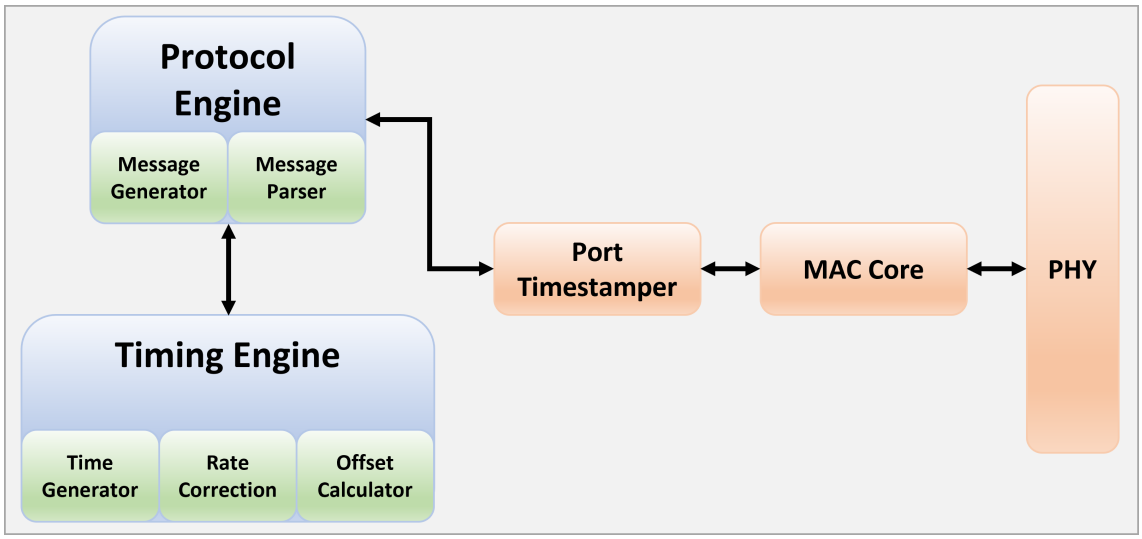

Figure 4.1 Proposed FPGA design of IEEE 1588 with rate correction

4.1.1 Message Generator Unit

It is the module where the necessary packets for the synchronization protocol are generated. This unit, which is managed by the signals from the Message Parser unit, generates Sync and Delay Response packets on the master node and the Delay Request packet on the slave node.

4.1.2 Message Parser Unit

This module parses and controls the type of packets received by the node. If a PTP packet is detected, the type of PTP packet is sent to the Message Generator unit and the time information read from the packet is transferred to the Offset Calculator unit.

4.1.3 Time Generator Unit

This unit is responsible for the generating time information of the PTP node. The counter is implemented as a 64-bit counter that consists of 32-bit seconds and 32-bit nanoseconds. From the start of the node's operation, the time counter begins to increment. Upon initialization of PTP on the master node, the value of the time counter is set to Universal Time Coordinated (UTC) by configuration and not permitted to change under any other circumstances. The value of the time information counter in the slave node is updated using the signals from the Offset Calculator unit and the Rate Correction unit.

4.1.4 Rate Correction Unit

In this module, the Rate Correction function explained in Section 3.3 is implemented. The signals generated in this module, which is active only in the slave node, are sent to the Time Generator Module. The operation of the protocol is not affected when the Rate Correction feature is switched between On and Off.

4.1.5 Offset Calculator Unit

This unit, where the time information transmitted by Ethernet packets is received and interpreted, is the main module that provides time synchronization. The module works only on the slave node to calculate the time difference. The time values obtained from the Message Parser are compared here to calculate the offset and the path delay. The calculated values are sent to the Time Generator module and time correction is performed. This unit also generates trigger signals, which are used to confirm synchronization precision and accuracy.

4.1.6 Port Timestamper Unit

This module basically consists of two parts as receiver and transmitter. In the transmitting part, PTP message is detected by checking the Ethernet packet and type of the PTP message is determined. While Sync and Delay Request messages are forwarded to the MAC Core, messages are stamped with time information, t_1 and t_3, respectively. If the transmitting message is Delay Response, timestamp t_4 is written inside the PTP message and sent.

In the receiving part, received Ethernet packet is parsed. When PTP message is detected, the type of the message is determined for timestamping operation. If the type of PTP message is Delay Response, time information in the message is received. If the type of message is Sync and Delay Request, message receiving time is recorded. This module is also placed just before the MAC Module to minimize processing delay and avoid asymmetry [34].

4.1.7 Mac Core Unit

The communication between the physical layer and PTP unit is provided by TEMAC IP Core that supplied from Xilinx. Ethernet core adds preamble and Frame Control Sequence (FCS) to the payload from the PTP unit via Advanced Extensible Interface (AXI). It also performs Media Access Control (MAC) address check for messages received from the physical layer via Reduced Gigabit Media Independent Interface (RGMII) and runs the Cyclic Redundancy Check (CRC) mechanism. The speed of the

Ethernet Mac core can be set to 10/100/1000 Mbps. In this thesis, it is operated as 1000 Mbps (1 Gbps).

4.2 Numerical Results

The test environment consists of two main components the End Systems and the Switches. PTP protocols runs on the End System as master or slave node. Switch provide communication between End Systems. BMCA defined for master-slave selection in the standard is not covered in this study. Instead of BMCA, one of the nodes is run as a master and the other as a slave. The tests are performed in 3 different topologies. Protocol packets are transmitted bidirectionally over one switch in the first topology, and two switches in the second topology. In the third topology, packets transmitted in the master-slave direction are forwarded over three switches, while packets transmitted in the slave-master direction are forwarded through two switches. The effect of the number of switches in the first two topologies and the physical asymmetry between two nodes in the third topology on the time synchronization are examined. The results are explained in terms of accuracy and precision. These terms are used for slave node time closeness to master node time and clarify to measured differences closeness, respectively.

Figure 4.2 shows an example test environment that prepared for Topology 3. This configuration uses three switches and three end systems to execute the PTP protocol.

Figure 4.2 Test environment example

End Systems are used to generate Ethernet traffic on the network while running PTP.

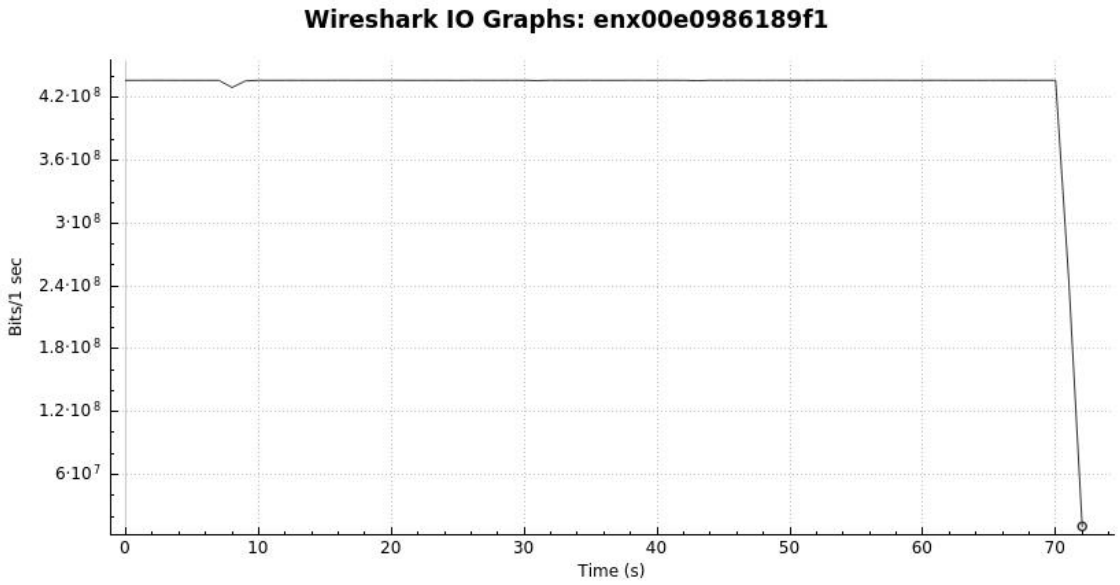

Figure 4.3 Ethernet traffic bandwidth usage graph

Ethernet traffic, 1500 bytes frames at 60 microseconds interval (approximately 200 Mpbs) is generated from each End System. There are two slave end systems sending Ethernet packets to the network, so the total Ethernet traffic is about 400 Mpbs. This ratio corresponds to approximately 40% of the total bandwidth. According to [38], the amount of traffic assumed to be used in a network set up for avionics is referenced. Switch(es) are configured to route these packets to End Systems. Ethernet traffic is captured with the Wireshark to measuring bandwith usage rate on the same PC used for debugging. Figure 4.3 shows the Ethernet background traffic rate in the network, which is generated in the slave nodes.

Figure 4.4 is an example where all packets transmitted and received between nodes are monitored on the PC over the switch. Protocol packets and debug packets are also seen in the figure.

In the IEEE 1588 standard, the synchronization interval is assumed to be 1 second. In addition, it is stated that it can be adjusted at certain intervals according to the requirement. Based on this flexibility of standard, the period of the Sync message is selected 125 milliseconds in order to be similar to the messaging interval recommended in the Time Sensitive Network (TSN) [39] standard.

The synchronization performance measurement is made with two debug packets that depend on receiving Sync packet is generated just before and after time correction. The first debug packet shows the maximum time difference between two nodes, and the second debug packet shows the minimum time difference. The interrupt signal created for the generation of packets in the slave node is sent to the master node via

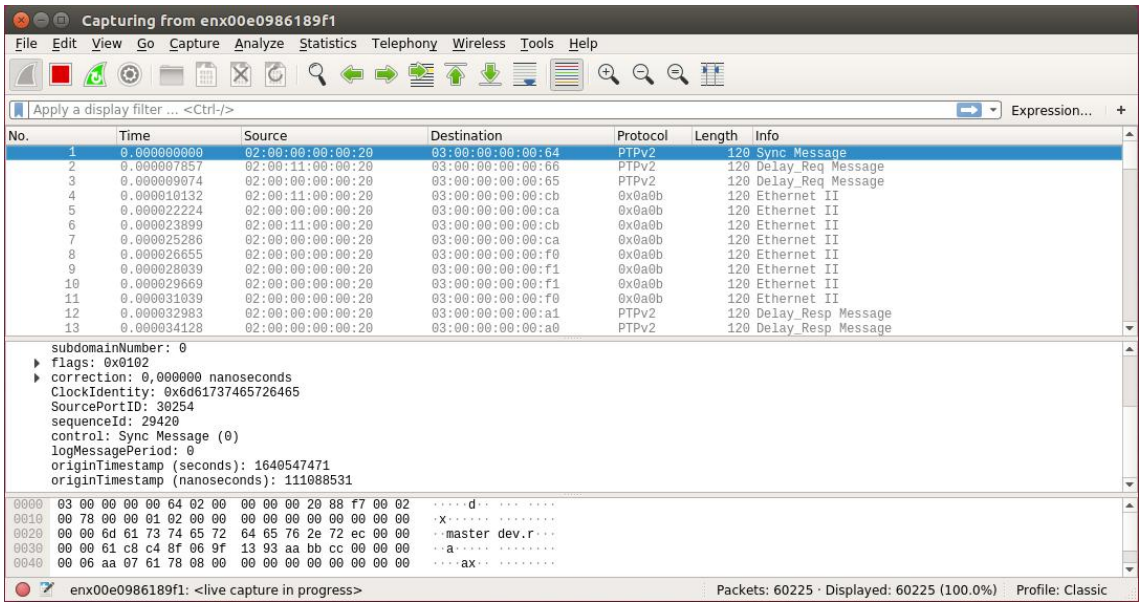

Figure 4.4 Example capture from Wireshark

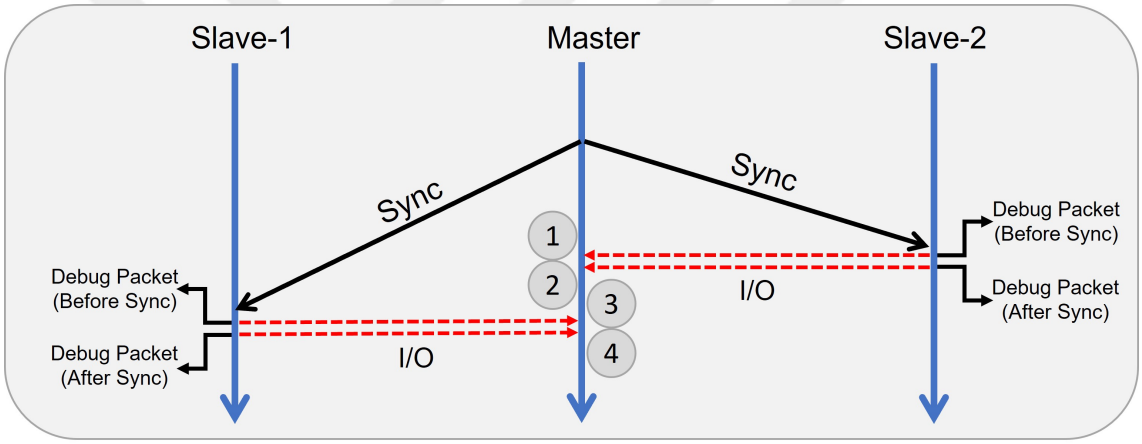

Figure 4.5 Debug process

I/O. In Debug packets, the sequence number of the Sync packet sent last from the master and received on the slave and the current time of the node are written. These Ethernet packets are transmitted to the PC via first switch. Python script developed by us is executed on a PC to compare the two debug packets and record the time differences. The method used for debugging is seen in Figure 4.5. For each scenarios, tests are carried out for 10 minutes and 9600 samples are taken.

4.2.1 Test Scenario 1 with Single Switch

The first topology results where a single switch provides the communication between master and slave nodes can be found in Table 4.2. Since this network is symmetric, the minimum time difference is measured as zero in all scenarios. In the absence of traffic, the maximum calculated time difference is the result that occurs because the nodes

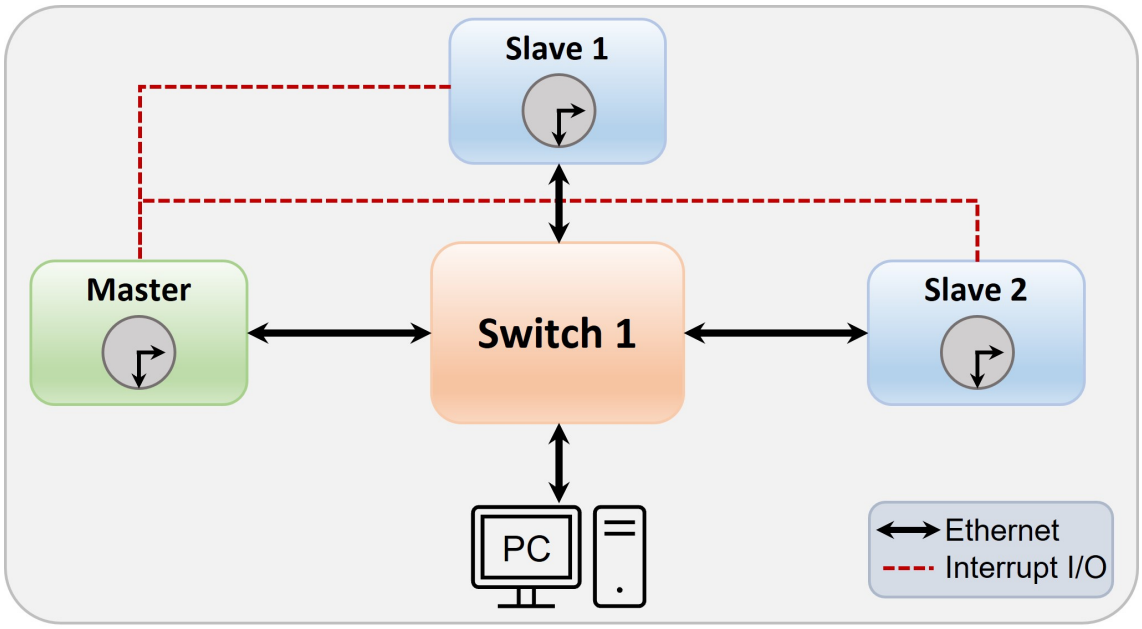

Figure 4.6 Topology 1 with single switch

are operating at different oscillators. When the frequency of synchronization packets are reduced, the time difference between nodes increases. The mean value of the time difference is calculated by adding the two values taken in each synchronization process and dividing by the number of samples. The average value is less than half of the maximum difference, which means that the time difference measured at the end of the synchronization process is close to zero. It has occurred that the times of slaves 1 and 2 are synchronized to master, but the maximum values are different due to the operation of its oscillators.

The presence of traffic in the network where synchronization is performed has two different effects. The first is the asymmetry in the path delay measurement. The second is the variation in the arrival interval of the synchronization packets to the slave node, which is required for the Rate Correction function. These effects result from congestion between protocol packets and other Ethernet packets forwarded to the same switch output port. No research or development is done in this study to correct this effect.

The scenario in which the Rate Correction is enabled and the synchronization interval is set to 125 milliseconds is the scenario in which the best synchronization accuracy is measured, regardless of Ethernet traffic. This is because frequent synchronization reduces the period in which the nodes generate the timing information themselves and enables faster correction of erroneous measurements.

According to results, if there is only PTP communicate between nodes and rate

Table 4.2 Topology 1 results

Rate Correction Status (On/Off)	Sync Interval (sec)	Traffic Case	Time Offset to Master (ns)			
			Slave 1		Slave 2	
			Mean	Max	Mean	Max
Off	1	Only PTP	3212	7216	3181	7224
		With Traffic	10167	27872	8805	25808
	0.125	Only PTP	588	1400	362	816
		With Traffic	7430	23080	3295	17592
On	1	Only PTP	238	1040	349	1056
		With Traffic	7358	38656	13078	42070
	0.125	Only PTP	239	560	322	576
		With Traffic	7090	26200	3666	29870

correction feature is disabled, synchronization accuracy is higher but precision is lower. Rate correction is enabled, precision is increased. While Ethernet background traffic is enabled and rate correction feature is disabled, accuracy is high but precision is low. If rate correction feature is enabled accuracy also decreases.

4.2.2 Test Scenario 2 with Double Switch

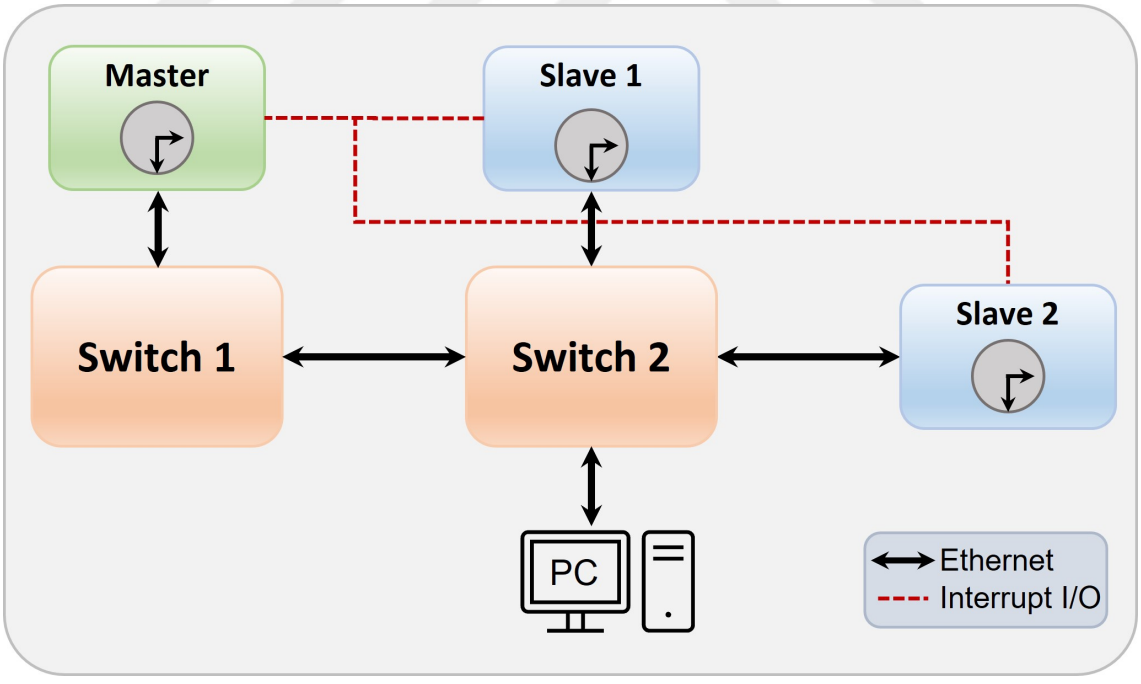

Figure 4.7 Topology 2 with double switch

Table 4.3 shows the results of the second topology, in which two switches provide communication between Master and Slave nodes. Packets are transmitted between nodes using two cascaded network switches in this network structure. In this scenario, it is desired to examine how the effect of the number of network switches between the

Table 4.3 Topology 2 results

Rate Correction Status (On/Off)	Sync Interval (sec)	Traffic Case	Time Offset to Master (ns)			
			Slave 1		Slave 2	
			Mean	Max	Mean	Max
Off	1	Only PTP	3370	7440	2796	6136
		With Traffic	19310	43024	15351	37080
	0.125	Only PTP	595	1432	375	840
		With Traffic	16988	30856	16414	30760
On	1	Only PTP	244	952	355	952
		With Traffic	22340	65216	20070	44616
	0.125	Only PTP	236	568	324	576
		With Traffic	16985	32712	17222	31880

nodes changes the synchronization precision. In a similar manner previous scenario, the best synchronization average and maximum difference values are measured when the Rate Correction feature is enabled, the protocol frequency is 125 milliseconds, and only the protocol packets are transmitted regardless of the number of switches.

The increase in the number of network switches causes protocol packets to arrive later. During this time, the time difference between the slave nodes, which produce their own time, and the master is greater than in the scenario with only one network switch. Similar to the previous scenario, rate correction feature increases the synchronization precision, while Ethernet background traffic reduces both of accuracy and precision.

4.2.3 Test Scenario 3 for Asymmetric Network

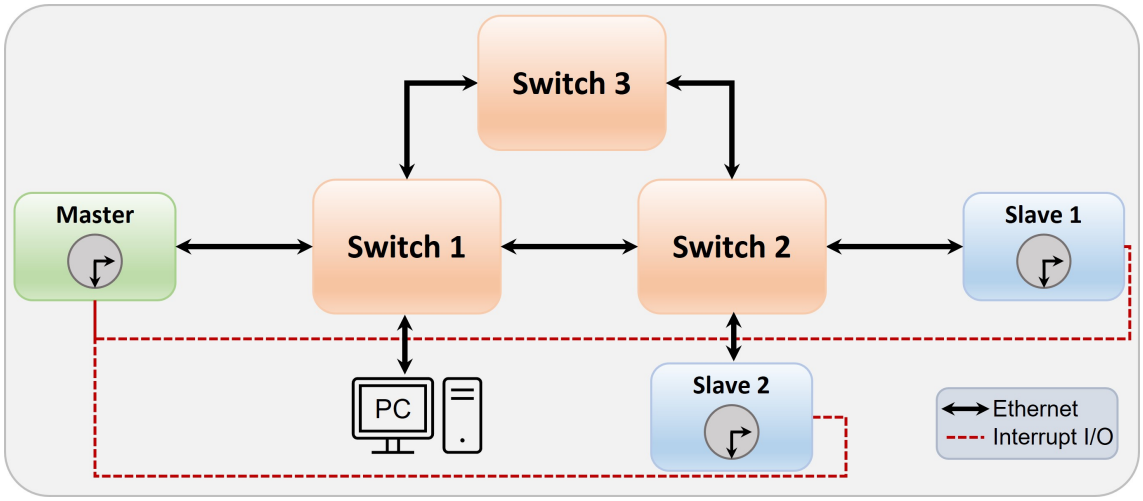

Figure 4.8 Topology 3 asymmetric network

The results of the third topology prepared to test the asymmetric path delay condition can be found in Table 4.4. In this topology prepared with three switches, packets from

Table 4.4 Topology 3 results

Rate Correction Status (On/Off)	Sync Interval (sec)	Traffic Case	Time Offset to Master (ns)			
			Slave 1		Slave 2	
			Mean	Max	Mean	Max
Off	1	Only PTP	2955	5392	4008	8128
		With Traffic	21318	33480	16288	27224
	0.125	Only PTP	892	1528	1542	2368
		With Traffic	18708	28072	17985	27312
On	1	Only PTP	1338	4808	1220	2072
		With Traffic	18432	27544	18368	27576
	0.125	Only PTP	1336	1672	1206	1688
		With Traffic	18254	27312	18298	27320

the master to the slave pass through three switches, while packets from the slave to the master pass through two switches. In all the scenarios performed in this topology, the synchronization accuracy is lower unlike the previous two topologies. In scenarios with Ethernet traffic, packets are transmitted across two network switches as they are generated on slave nodes. Therefore, packets are transmitted from the slave to the master traverse fewer switches; they are arrived later due to the congestion on the switch. This situation reduces the physical asymmetry to some extent and enhances synchronization accuracy. The use of three switches in the transmission of the Sync packet causes the path delay value to be miscalculated and higher than the actual value. Therefore, a one-to-one comparison of the results of this topology with the previous one does not provide a correct explanation. However, the synchronization levels around 100 microseconds despite all these effects.

5
RESULTS AND DISCUSSION

The next-generation aircraft have advanced features thanks to the latest technology products such as radar, camera, and weapon systems. An avionics architecture called IMA is used to make the systems in which these products are used more flexible and reliable. In the IMA architecture, high-rated data from other units are transmitted over the avionics data network to central computers to generate useful information for the user and the system. For the information generated from the data processed in the central computer to be valid, the data sources should have a common notion of time. Therefore, time synchronization of the avionic sub-units in the IMA architecture is an essential requirement. In this work, the IEEE 1588 protocol has been implemented to solve the need for time synchronization of the avionics units without overloading the existing IMA architecture with new tasks.

The FPGA development boards prepared to represent the avionic units, which are running PTP by using own FPGA, are called the end systems. In this study, time synchronization protocol runs for end systems, and switches connect the end systems. A new feature called Rate Correction is proposed to improve the performance of PTP and increase its precision.

The time difference between nodes is provided by the delay request- response mechanism used in PTP. Even if times are the same after completion of the protocol, time synchronization is disrupted due to the frequency drift between the nodes until the following synchronization sequence occurs. Rate Correction function is used to correct the frequency drift which is the cause of time difference during synchronization intervals.

During testing, the precision of time synchronization is reported when Ethernet traffic and the Rate Correction feature are active in the network where PTP is running. Although the path delay of packets during synchronization in PTP is considered symmetric, it is measured to be asymmetric due to congestion on the switches used to transmit the packets, which reduces the precision of synchronization. The results

also demonstrate that the Rate Correction feature improves synchronization accuracy. The tests and results regarding the synchronization precision of PTP on development board are described in Chapter 4.

This thesis suggests that PTP can be used to synchronize avionics systems time as it is Ethernet- based, provides high synchronization precision, and is easy to use. However, performed tests prove that the precision decreases when the path delay is considered symmetric, and the frequencies are shifted between the units in PTP. Therefore, PTP is a protocol that still has the necessity for development in these aspects. We plan to make improvements in these areas in our future work.

REFERENCES

[1] G. M. A. Shah, "Avionics modification research analysis: From electromechanical to digital avionics and from digital to integrated modular avionics (ima)," M.S. thesis, Universitat Politècnica de Catalunya, 2014.

[2] G. T. Lemons, K. Carrington, "F-35 mission systems design, development & verification," in *2018 Aviation Technology, Integration, and Operations Conference*, 2018, p. 3519.

[3] E. Blasch, P. Kostek, P. Pačes, K. Kramer, "Summary of avionics technologies," *IEEE Aerospace and Electronic Systems Magazine*, vol. 30, no. 9, pp. 6–11, 2015.

[4] P. Bieber, F. Boniol, M. Boyer, E. Noulard, C. Pagetti, "New challenges for future avionic architectures.," *AerospaceLab*, no. 4, p–1, 2012.

[5] T. Schuster, D. Verma, "Networking concepts comparison for avionics architecture," in *2008 IEEE/AIAA 27th Digital Avionics Systems Conference*, IEEE, 2008, pp. 1–D.

[6] C. M. Fuchs, A. S. Schneele, "The evolution of avionics networks from arinc 429 to afdx," *Innovative Internet Technologies and Mobile Communications (IITM), and Aerospace Networks (AN)*, vol. 65, pp. 1551–3203, 2012.

[7] C. B. Watkins, R. Walter, "Transitioning from federated avionics architectures to integrated modular avionics," in *2007 IEEE/AIAA 26th Digital Avionics Systems Conference*, IEEE, 2007, 2–A.

[8] R. Black, M. Fletcher, "Next generation space avionics: A highly reliable layered system implementation," in *The 23rd Digital Avionics Systems Conference (IEEE Cat. No. 04CH37576)*, IEEE, vol. 2, 2004, 13–E.

[9] Z. Idrees, J. Granados, Y. Sun, S. Latif, L. Gong, Z. Zou, L. Zheng, "Ieee 1588 for clock synchronization in industrial iot and related applications: A review on contributing technologies, protocols and enhancement methodologies," *IEEE Access*, vol. 8, pp. 155 660–155 678, 2020.

[10] K. Harris, "An application of ieee 1588 to industrial automation," in *2008 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication*, IEEE, 2008, pp. 71–76.

[11] M. Faizullin, A. Kornilova, G. Ferrer, *Open-source lidar time synchronization system by mimicking gps-clock*, 2021. arXiv: 2107.02625 [cs.RO].

[12] J. C. Eidson, M. Fischer, J. White, "Ieee-1588™ standard for a precision clock synchronization protocol for networked measurement and control systems," in *Proceedings of the 34th Annual Precise Time and Time Interval Systems and Applications Meeting*, 2002, pp. 243–254.

[13] F. Boulanger, D. Marcadet, M. Rayrole, S. Taha, B. Valiron, "A time synchronization protocol for a664-p7," in *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, IEEE, 2018, pp. 1–9.

[14] S. Rinaldi, D. Della Giustina, P. Ferrari, A. Flammini, E. Sisinni, "Time synchronization over heterogeneous network for smart grid application: Design and characterization of a real case," *Ad Hoc Networks*, vol. 50, pp. 41–57, 2016.

[15] F. Sivrikaya, B. Yener, "Time synchronization in sensor networks: A survey," *IEEE network*, vol. 18, no. 4, pp. 45–50, 2004.

[16] P. Park, M. Son, J. Lee, J. Yoon, "Performance evaluation of the efficient precise time synchronization protocol for the redundant ring topology network," in *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*, IEEE, 2021, pp. 1–10.

[17] H. Guo, P. Crossley, "Design of a time synchronization system based on gps and ieee 1588 for transmission substations," *IEEE Transactions on power delivery*, vol. 32, no. 4, pp. 2091–2100, 2016.

[18] S. J. Kappan, V. Vivekanand, "Avionic data acquisition system using mil std 1553b controller with irig-b timecode decoder," in *2014 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2014]*, IEEE, 2014, pp. 1127–1133.

[19] K. Behrendt, K. Fodero, *et al.*, "The perfect time: An examination of time-synchronization techniques," *Publication, Schweitzer Engineering Laboratories, Inc*, pp. 1–18, 2006.

[20] S. Johannessen, "Time synchronization in a local area network," *IEEE control systems Magazine*, vol. 24, no. 2, pp. 61–69, 2004.

[21] M. Glass, "Buses and networks for contemporary avionics," *SAE Transactions*, pp. 802–818, 2007.

[22] M. A. Lombardi, "Microsecond accuracy at multiple sites: Is it possible without gps?" *IEEE Instrumentation & Measurement Magazine*, vol. 15, no. 5, pp. 14–21, 2012.

[23] D. M. Ingram, P. Schaub, D. A. Campbell, R. R. Taylor, "Evaluation of precision time synchronisation methods for substation applications," in *2012 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication Proceedings*, IEEE, 2012, pp. 1–6.

[24] M. Lipiński, T. Włostowski, J. Serrano, P. Alvarez, "White rabbit: A ptp application for robust sub-nanosecond synchronization," in *2011 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication*, IEEE, 2011, pp. 25–30.

[25] "Ieee standard for a precision clock synchronization protocol for networked measurement and control systems," *IEEE Std 1588-2008 (Revision of IEEE Std 1588-2002)*, pp. 1–269, 2008. DOI: 10.1109/IEEESTD.2008.4579760.

[26] S. B. Moon, P. Skelly, D. Towsley, "Estimation and removal of clock skew from network delay measurements," in *IEEE INFOCOM'99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now (Cat. No. 99CH36320)*, IEEE, vol. 1, 1999, pp. 227–234.

[27] K. Correll, N. Barendt, M. Branicky, "Design considerations for software only implementations of the ieee 1588 precision time protocol," in *Conference on IEEE*, Citeseer, vol. 1588, 2005, pp. 11–15.

[28] L. Benetazzo, C. Narduzzi, M. Stellini, "Analysis of clock tracking performances for a software-only ieee 1588 implementation," in *2007 IEEE Instrumentation & Measurement Technology Conference IMTC 2007*, IEEE, 2007, pp. 1–6.

[29] B. Zhao, N. Wang, "The implementation of ieee 1588 clock synchronization system based on fpga," in *Fifth International Conference on Intelligent Control and Information Processing*, IEEE, 2014, pp. 216–220.

[30] Y. Lin, L. Hao, T. Dan, "Research and implementation in synchronized system of data acquisition based on ieee 1588," in *IEEE 2011 10th International Conference on Electronic Measurement & Instruments*, IEEE, vol. 2, 2011, pp. 198–201.

[31] W. Wallner, "Simulation of the ieee 1588 precision time protocol in omnet++," *arXiv preprint arXiv:1609.06771*, 2016.

[32] Y. Zhao, Y. Wang, J. Huang, X. Shi, "A stable clock synchronization based on clock drift rate," in *2008 IFIP International Conference on Network and Parallel Computing*, IEEE, 2008, pp. 204–209.

[33] S. Meier, H. Weibel, K. Weber, "Ieee 1588 syntonization and synchronization functions completely realized in hardware," in *2008 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication*, IEEE, 2008, pp. 1–4.

[34] D. Pedretti, M. Bellato, R. Isocrate, A. Bergnoli, R. Brugnera, D. Corti, F. Dal Corso, G. Galet, A. Garfagnini, A. Giaz, *et al.*, "Nanoseconds timing system based on ieee 1588 fpga implementation," *IEEE Transactions on Nuclear Science*, vol. 66, no. 7, pp. 1151–1158, 2019.

[35] W. Alghamdi, M. Schukat, "Advanced methodologies to deter internal attacks in ptp time synchronization networks," in *2017 28th Irish Signals and Systems Conference (ISSC)*, IEEE, 2017, pp. 1–6.

[36] G. Cena, I. C. Bertolotti, S. Scanzio, A. Valenzano, C. Zunino, "Synchronize your watches: Part ii: Special-purpose solutions for distributed real-time control," *IEEE Industrial Electronics Magazine*, vol. 7, no. 2, pp. 27–39, 2013.

[37] H. Liu, J. Liu, T. Bi, J. Li, W. Yang, D. Zhang, "Performance analysis of time synchronization precision of ptp in smart substations," in *2015 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication (ISPCS)*, IEEE, 2015, pp. 37–42.

[38] A. Finzi, A. Mifdaoui, F. Frances, E. Lochin, "Incorporating tsn/bls in afdx for mixed-criticality applications: Model and timing analysis," in *2018 14th IEEE International Workshop on Factory Communication Systems (WFCS)*, IEEE, 2018, pp. 1–10.

[39] I. S. Association *et al.*, "Ieee std 802.1 as-2011, ieee standard for local and metropolitan area networks—timing and synchronization for time-sensitive applications in bridged local area networks," *Mar*, vol. 30, p. 292, 2011.

PUBLICATIONS FROM THE THESIS

Conference Papers

1. M. E. Yabacioglu, M. S. Demir, S. N. Engin, et al., “Fpga implementation of precision time protocol and its performance analysis under traffic load,” in 2021 29th Signal Processing and Communications Applications Conference (SIU), IEEE, 2021, pp. 1–4.