Extending the precision time protocol to a metropolitan area network

Synchronizing radio base stations

MOZHDEH KAMEL

KTH Information and
Communication Technology

Degree project in
Communication Systems
Second level, 30.0 HEC
Stockholm, Sweden

Extending the precision time protocol to a metropolitan area network

Synchronizing radio base stations

Mozhdeh Kamel

<mozhdeh@kth.se>

2014.02.06

Master thesis

Examiner: Gerald Q. Maguire Jr.
Supervisors: Jiang Wu

School of Information and Communication Technologies
KTH Royal Institute of Technology
Stockholm, Sweden

Abstract

When building various types of wide area cellular radio networks there is a need to synchronize all of the base stations within a given system. Today this is typically done by attaching a highly accurate clock to each radio base station. A GPS radio receiver is commonly used as such a clock. This thesis explores the use of the Precision Time Protocol (PTP) to provide synchronization of radio base stations, rather than the current practice of using GPS radio receivers.

Advantages of utilizing PTP rather than a GPS radio receiver include the ability to easily locate radio base stations (without the need for connecting the GPS radio receiver to an antenna that has line of sight to a sufficient number of GPS satellites); the system is not vulnerable to interference with or jamming of GPS radio signals; the system is not vulnerable to spoofing of GPS radio signals, and because the new generations of radio base stations are connected to a packet based backhaul link – the system can potentially utilize the existing packet network interface (thus avoiding the need for a serial interface to the GPS receiver and a pulse per second input).

At the start of this thesis project it was not known what the limits of PTP are (in terms of utilizing PTP together with radio base stations). Thus it was not clear whether PTP could be extended to much longer distances than it had originally been designed for.

This thesis shows that PTP can be used as an accurate timing source to synchronize base stations in networks with up to four switches between the PTP grandmaster and any PTP slave.

This project was performed in the Common Transport Feature department at Ericsson.

Keywords: precision time protocol, radio base station, synchronization

Sammanfattning

Vid konstruktion av wide area cellular radio networks finns det behov av att synkronisera samtliga basstationer inom ett givet system. Detta görs idag typiskt genom att ansluta en klocka med stor tillförlitlighet till varje basstation. En GPS radiomottagare används vanligen som klocka för detta syfte. Detta examensarbete undersöker användandet av Precisions Tid Protokoll (PTP) för att synkronisera radiobasstationer, istället för att som nu typiskt använda GPS radiomottagare.

Fördelar med att använda PTP istället för GPS radiomottagare är att en radiobasstation lätt kan lokaliseras (utan att ansluta en GPS-mottagare till en antenn vilken har mottagning mot flera GPS-satelliter); systemet är inte sårbart mot interferens eller störningar av GPS radio signaler; systemet är inte sårbart mot spoofing av GPS radio signaler och på grund av att den nya generationens radiobasstationer är anslutna till ett paketförmedlande backhaul nätverk kan systemet potentiellt använda sig av det redan existerande paketförmedlande nätverksgränssnittet (och på sätt undvika ett seriellt gränssnitt mot en GPS-mottagare och en puls per sekund ingång).

När detta examensarbete startades var det inte känt var gränserna för PTP låg när det gäller att använda PTP tillsammans med radiobasstationer. Det var således inte klart ifall räckvidden för PTP kunde utvidgas till mycket längre avstånd än det ursprungligen var ämnat för. Detta examensarbete syftar till att visa att PTP kan användas som tillräckligt noggrann synkroniseringskälla för basstationer i nätverk med upp till fyra nätverksswitchar mellan PTP Grand Master och PTP slav.

Examensarbetet har utförts vid avdelning Common Transport Feature på Ericsson.

Nyckelord: precisions tid protokoll, radiobasstation, synkronisering

Acknowledgement

I would like to express my gratitude to my academic supervisor and examiner at KTH, Professor **Gerald Q. Maguire Jr.** for guiding me during this thesis project, providing helpful feedback and ideas during this thesis project.

I also would like to thank my industrial supervisor and my manager in Ericsson **Jiang Wu** and **Sheng Chu Li** and who gave the opportunity to do this thesis at Ericsson AB. I would also like to thank **Mikael Johansson** and **Mikael Olofsson** in the Network Synchronization department for explanation about PTP protocol and current synchronization technologies. Moreover, I like to thank to **Seth Norgren**, **Thomas Bäverlid** and **Goran Matovic** in CTF lab for lending me necessary equipment for this thesis process.

Furthermore, I like to special thanks to my friend **Pooya Moazzeni Bikani** for supporting me spiritually throughout my thesis.

Last but not least, many thank to my parents and my sisters for their unconditional support and love throughout my life.

Table of Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

Advanced LTE	Advanced Long Term Evolution
AOCM	Adaptive Oscillator Correction Method
BC	Boundary Clock
BMC	Best Master Clock
CDMA	Code Division Multiple Access
CPU	Central Processing Unit
FDD	Frequency Division Duplexing
FIFO	First In First Out
FPGA	Field-Programmable Gate Array
GB	Ggigabyte
GHz	Ggigahertz
GM	Grandmaster
GPS	Global Positioning System
GSM	Global System for Mobile
LAN	Local Area Network
LTE	Long Term Evolution
MAC	Media Access Control
MAN	Metropolitan Area Network
Mbps	Megabits per second
MIB	Management Information Base
minTDEV	Minimum Time Deviation
ms	Millisecond
NA	Not Applicable
NS	Nanosecond
NTP	Network Time Protocol
OC	Ordinary Clock
OCXO	Oven-Controlled Crystal Oscillator
OS	Operating System
PCI	Peripheral Component Interconnect.
PLL	Phased Locked Loop
PPB	Pulse per Billion
PPM	Pulse per Million
PPS	Pulse per Second
PQ	Priority Queuing
PRC	Primary Reference Clock
PIM-SM	Protocol Independent Multicast-Sparse Mode
PTP	Precision Time Protocol
PTPD	Precision Time Protocol Daemon
QoS	Quality of Service
RBS	Radio Base Station
RTC	Real-Time Clock
RP	Rendezvous Point
SNMP	Simple Network Management Protocol
Sync-E	Synchronous Ethernet
TC	Traffic Control
TC	Transparent Clock

TDD	Time Division Duplexing
TDev	Time Deviation
TDM	Time Division Multiplexing
TLV	Type-Length-Value
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System
UTC	Coordinated Universal Time
WAN	Wide Area Network
WCDMA	Wideband Code Division Multiple Access
WiMAX	Worldwide Interoperability for Microwave Access
2G	second generation

1 Introduction

This chapter provides a general introduction to this thesis. First, an overview of the area will be given, followed by the problem description, initial goals, limitations, and a summary of the thesis structure.

1.1 Overview

When building various types of wide area cellular radio networks there is a need to synchronize all of the base stations within a given system. Today this is typically done by attaching a highly accurate clock to each radio base station (RBS). In the field this is commonly done using a global positioning system (GPS) radio receiver to provide both the time of day and to discipline an oscillator. In traditional circuit switched telephony synchronization is done with an atomic clock, to which all of the elements of the operator's entire network are synchronized.

In packet switched networks two protocols have been used for synchronization: the network time protocol (NTP) and the precision time protocol (PTP). NTP has been widely used to synchronize computers to external time sources (such as one or more atomic clocks) or internal time sources (such as those using quartz oscillators). NTP has been successfully used together with local, metropolitan, regional, and even national networks (where the one-way delay is below 128 millisecond (ms)). However, NTP is limited to an accuracy of about 1 ms when used in a metropolitan area network (MAN); unless an accurate one pulse per second source is available (such as from a GPS receiver), in which case it is possible to achieve an accuracy of ~50 microsecond (µs). PTP, described in the IEEE 1588 standards, has been successfully used to synchronize clocks within a local area network with an absolute timing accuracy of better than 100 nanoseconds (ns) with respect to Coordinated Universal Time (UTC).

1.2 Goals

Figure 1-1 shows an overview of the planned use of PTP over a MAN. In this system timing information (1 pulse per second (PPS) and time-of-day) are provided by a grandmaster (GM) clock. The actual grandmaster clock might be a GPS receiver or an atomic clock. This GM clock acts as a master for all of the clocks *within* this network. A RBS which needs to be synchronized acts a slave clock at the edge of the MAN and receives timing information via its network interface (for example an Ethernet interface). The RBS will use this information to discipline its local clock.

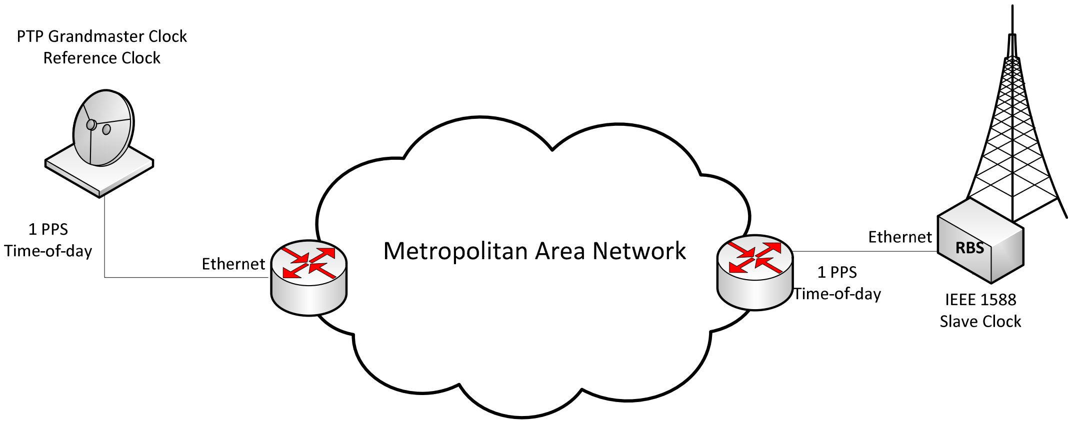

Figure 1-1: **PTP over MAN**

Different types of RBS (e.g., GSM, Wideband Code Division Multiple Access (WCDMA), and Long Term Evolution (LTE)) have different timing requirements (these requirements will be described in detail in section 2.1.1). While PTP can normally meet these timing requirements in a local area network (LAN), in this thesis project we will examine whether PTP can meet these requirements in a MAN. To do this we will explicitly calculate a delay budget and examine the expected jitter in a MAN to which the various RBSs are attached. The main scientific question of this thesis project is to understand how well PTP can keep the slave within a bounded time of the GM and for what period of time the slave is within this bound.

1.3 Limitations

This section states the limitations of this thesis with respect to the study and implementation that have been carried out within this project.

Initially, the planned thesis was to include a comparison between a simulation of the PTP protocol and experimental results using real hardware. However, after several months of working with a PTP implementation in OPNET, this effort was terminated because some parts of the code were missing and unavailable from the original author of the code.

Following this the thesis aim shifted to an experiment with both full time aware networks and partial time aware networks. Unfortunately, budget limitations prevented the acquisition of a real hardware boundary clock (BC) or the emulation of a BC.

As a result the experiments were done with older borrowed PTP devices that do not support PTPv2 (which is currently used in market). Moreover, those devices only support a special kind of GPS receiver. As a result the experiments were limited to measuring the time difference between a GM and slave where the GM stratum is 255 instead of 1. This means that the GM did not have a GPS disciplined oscillator and this GM was *not* synchronized with UTC; the results is that it was only possible to measure the *relative* deviation of the slave from this GM, rather than the *absolute* deviation from UTC.

A further limitation of this thesis was that the industrial team, I was working with, did not have any competence in the area of synchronization; hence I have done this thesis completely independently. Having a larger budget and support from team mates, who had competence in synchronization would have been helpful and reduced the time necessary to carry out this thesis project.

1.4 Methodology

This section describes the thesis framework used to achieve the thesis project's goals (as were described in section 1.2). A quantitative research approach was chosen for this thesis due to the desire for quantitative values for the *maximum* delay variation. Qualitative research methods were ruled out as a goal of the project was to determine **if** PTP can meet these timing requirements in a MAN. If PTP is able to meet the requirements described in section 2.1, then industry *could adopt* PTP for RBS synchronization – eliminating the need for each RBS to be connected to a GPS receiver.

1.4.1 Literature Study

The first step in this thesis project was to study the timing requirements for different RBS technologies. Next the standards for synchronization in packet networks were studied. This provided the essential background knowledge regarding synchronization in packet switched network. Two major synchronization methods were considered: NTP and PTP. Given the requirements described in section 2.1, the main focus was on PTP. Section 2.3 summarizes the most relevant research regarding PTP (as defined in IEEE 1588).

1.4.2 Experiment Design

In this thesis project several scenarios were considered in order to evaluate the effects of the characteristics of a large network on the PTP protocol. The two main network characteristics considered were: the number of nodes in the network and the load on the network.

This experimental study is based on the deductive research approach[1]. A full description of how the experiment was designed and carried out will be described in Chapter 3.

1.4.3 Data Collection and Analysis

This thesis project is measurement oriented, which resulted in the selection of a quantitative research method. Data was collected from an isolated network that was created for each of the scenarios (details of these scenarios are given in section 4.3). A digital oscilloscope was used to compare the *relative* time difference between a slave and the GM. Each test was repeated two times to decrease the confidence interval of the result, i.e., to increase our confidence that the time deviation between the slave and the GM is within the requirements of the *most* demanding RBS technology. The experimental setup used when collecting data is described in section 3.4.2. The analysis of the collected data will be presented in Chapter 5.

1.4.4 Conclusion

The result of measurements carried out in this thesis project suggests that PTP can be used in a small MAN, specifically a MAN with a maximum of four nodes between the GM and any slave. Note that this thesis project has only shown that this suggestion is valid when the GM is **not** connected to a primary reference clock (PRC), such as a GPS receiver or atomic clock. The evaluation with a GM connected to a PRC is left as future work. It worth mentioning we expect increasing the accuracy by connecting GM to a PRC.

1.5 Structure of this thesis

The first chapter of this thesis has presented the problem that is to be solved. Chapter 2 provides the reader with the necessary background to read the rest of the thesis. Chapter 2 also summarizes related work that is directly relevant to this thesis. Chapter 3 describes the method and methodology used. Chapter 4 describes the design and implementation of a prototype solution. Chapter 5 evaluates this prototype with the goal of determining the bounds of operation for PTP with respect to the requirements for synchronization of different types of RBSs. Chapter 6 summarizes the conclusions that can be drawn from this thesis project, proposes future work, and offers reflections on some of the economic, social, and ethical considerations relevant to this thesis project.

2 Background and related work

This chapter provides the background information necessary for the reader to understand the rest of this thesis. It begins with the requirements for synchronization of different types of RBS, and then describes some of the technologies that will be used in our proposed solution (starting with NTP and PTP). The chapter also describes how PTP needs to be addressed by the different elements of the network. The chapter concludes with a summary of related research.

2.1 Requirements for base station synchronization

A frequency division duplexing (FDD) network uses two separate sets of frequencies for uplink and downlink. In order to support the handoff process in different FDD radio technologies, the proper frequency must be used. In all radio standards that use this technique, both the mobile handset and the RBS must use the same frequency. FDD is used by Global System for Mobile (GSM), Universal Mobile Telecommunication System (UMTS) Wideband Code Division Multiple Access (WCMDA), Long Term Evolution (LTE) FDD, and Worldwide Interoperability for Microwave access (WiMAX) FDD [2].

Unlike FDD, time division duplexing (TDD) uses the same frequency, but different time slots for uplink and downlink. Therefore in these networks, in addition to using the correct frequency, the mobile device and the RBS time must also know when the timeslot occurs [2].

The requirements for synchronization for various radio standards are shown in Table 2-1.

Table 2-1: Synchronization accuracy required of various radio standards (inspired by the table in [3])

Radio Standard	Frequency Accuracy	Time Accuracy
CDMA2000[3]	50 ppb	±3µs (should) ±10 µs (shall)
GSM [4]	50 ppb	NA[*]
UMTS WCDMA FDD[5]	50 ppb	NA
UMTS WCDMA TDD [6]	50 ppb	±2.5µs
UMTS WCDMA Femtoc[3]	250 ppb	NA
TD SCDMA[3]	50 ppb	±3µs
LTE FDD[3]	50 ppb	NA
LTE TDD[3]	50 ppb	±1.5µs small cells, ±0.5µs large cells
WiMAX TDD[3]	3 ppm absolute 50 ppb between base stations	±1-8 µs (depending on implementation)

---

[*] NA stands for Not applicable.

2.1.1 Base station synchronization today

In cellular radio networks, synchronization is necessary in different parts of the network, specifically the network itself, the nodes of the network, and the radio interface(s) attached to these network nodes. Moreover, there are different kinds of clock synchronization in the RBSs. Frequency synchronization is necessary to support the call handover mechanism and because the base stations need to stay within their allocated spectrum (this prevents them from interfering with other cells). Phase/time synchronization is necessary for base stations that use single frequency techniques, such as TDD, in order to ensure that all of the relevant clocks are aligned in phase [7].

Traditional mobile backhaul (used to connect the RBS to the operator's core network) used a circuit switched network, typically time division multiplexing (TDM) links. In these systems all elements of an operator's network were synchronized to a single clock. As noted in the previous section different types of RBS need different degrees of synchronization. Circuit-switched networks need time synchronization, but packet switched networks are not strict in their need for time synchronization [3]. Hence based on the backhaul network used the RBS will (in the case of a circuit switched backhaul link) or will not have an available synchronization source via the backhaul link (in the case of a packet switched backhaul link). However, due to the increasing demand for greater link bandwidths, today mobile backhaul is transitioning to packet based transport networks based upon IP/MPLS and carrier Ethernet.

2.1.1.1 Synchronization in 2G/GSM

In second generation (2G) mobile networks, such as GSM, the RBS need frequency synchronization. Reference clocks are distributed around the network and are used to provide the needed frequency synchronization [8]. The frequency accuracy required by GSM was shown in Table 2-1.

2.1.1.2 Synchronization in TD-SCDMA

In TD-SCDMA, the RBS needs both frequency and time synchronization [8]. The synchronization requirements for TD-SCDMA were shown in Table 2-1.

2.1.1.3 Synchronization in WCDMA

WCDMA does not rely on time-of-day information, but frequency synchronization is needed [3]. The synchronization requirements for the different forms of WCDMA were shown in Table 2-1.

2.1.1.4 Synchronization in WIMAX

In addition to frequency synchronization, WiMAX needs time synchronization to ensure that all base stations switch between uplink/downlink for TDD access [3]. The synchronization requirements were shown in Table 2-1.

2.1.1.5 Synchronization in LTE/Advanced LTE

LTE requires 100 megabits per second (Mbps) and faster backhaul transport networks, so TDM links may not be suitable for this backhaul. In LTE technology frequency synchronization is needed to keep each base station within its assigned spectrum, to ensure phase synchronization, and to provide the time-of-day [7].

2.1.1.6 RBS synchronization summary

To support synchronization among distributed time references, physical synchronization such as Synchronous Ethernet (Sync-E) or packet based synchronization (such as IEEE1588 or NTP) may be used. In WiMAX-TDD, TD-CDMA, and TD-SCDMA RBS, a GPS receiver is the current preferred method of synchronization [4]. However, a recent master's thesis by Elham Khorami [9] suggests that chip scale atomic clocks may be cost effective replacements for internal crystal oscillators. Such a chip scale atomic clock would provide a signal that could be used for frequency control, but there is still a need for setting the time of day and synchronization of multiple chip scale atomic clocks.

2.1.2 Base station synchronization in the near future

The most recent base stations that are introduced to market by vendors are micro, pico, and femto base stations. The intent of introducing such base stations is to increase coverage in indoor areas, such as a small campus building or residential building or house in which the signal penetration is weak.

Micro and pico base stations are suitable for a small campus and individual buildings. These base stations use an oven-controlled crystal oscillator (OCXO) with an ultra stable phase lock loop (PLL) and support packet access to a high data rate backhaul. They are frequently connected via a Metro Ethernet Network which may be suitable for PTP [10].

Femto cells are low power base stations that provide shorter range wireless links which makes them suitable for deployment in a weak signal environment, such as small house [11]. They use a temperature compensated crystal oscillator (TCXO) and PLL which is moderately stable. As both pico and femto base stations cover indoor areas, this shorter range requires them to have a frequency accuracy of up to 250 ppb at the RF output [12], hence the input reference should be accurate at approximately ±0.1 ppm [13].

2.2 Network time protocol

NTP [2] is a widely used time synchronization protocol. NTP was developed over 20 years ago (in 1985) [3]. NTP provides synchronization in a hierarchical fashion between servers and clients. This protocol uses the user datagram protocol (UDP) for communication between clients and servers [11]. Each hierarchical time synchronization structure contains clients, servers, and peers. A client is a clock that queries for the current time from one or more servers as its reference (for true time). A server is a clock that provides a time reference for clients. Peers compare their own time with other peers in order to synchronize their sense of time.

2.2.1 NTP's clock stratum

In NTP, the different hierarchical layers define different stratum. NTP defines 256 strata, but stratum 16 is considered to be unsynchronized. Three of the most important strata are:

Stratum 0 is a reference clock which is considered the most accurate clock [14].
Stratum 1 is a server that directly connected [14].
Stratum 2 is a server that connects to stratum 1 [14].

2.2.2 NTP versions

NTP exists in different versions and is capable of providing synchronization within milliseconds. The current version is NTP [11] version 4 which provides an accuracy of milliseconds over a wide area network (WAN), sub millisecond accuracy over a LAN, and sub microsecond accuracy when using a cesium oscillator as its precise time resource [13].

2.2.3 Simple Network Time Protocol (SNTP)

As NTP has a complicated implementation, hence it is not suitable for all systems. SNTP is a simplified version of NTP. However, this simplicity results in lower quality time synchronization compared with a full NTP implementation. Moreover, different methods are used in NTP and SNTP for checking timing errors and correcting the time. NTP utilizes a more complicated algorithm and uses several servers to estimate the current time. Unlike NTP, SNTP uses a single Ethernet connected time server as a reference for time, but if the primary server is unavailable it can utilize a backup server [14].

2.3 Precision time protocol

PTP is described in IEEE 1588 [14]. PTP was designed for precise synchronization among different distributed network nodes [15]. Synchronization is achieved in this protocol by utilizing hierarchical master and slave clocks. Each node can be a master for nodes which have a lesser clock and will become a slave to a node which has a better clock. PTP was standardized in two versions: version one was published in 2002 and version two (an improvement of version one) was published in 2008. Macé and van Kempen describe the differences between these two versions in [20]. In the following text we will refer to these two versions as PTPv1 and PTPv2.

2.3.1 Difference between PTPv1 and PTPv2

PTPv1 was designed to achieve the following objectives: to enable synchronization to an accuracy of at least sub-microseconds of real time clocks in network nodes, to be suitable for a LAN that supports multicast, applicable in heterogeneous networks that have different clock accuracies, and applicable in a network with minimum resources or without any administration [21]. This protocol was also designed to be simple and involve minimal host components [22].

PTPv2 was published six years later and added some improvements, such as increased efficiency in comparison to the previous version. It was designed to achieve synchronization better than 1 nanosecond, support unicast rather that multicast, work over asymmetrical communication link, added transparent clocks, provide redundancy, provide better sampling and message rate, added a mechanism for message extensions, provide configuration of hierarchical synchronization, separated synchronization messages from signalling messages, and maps to UDP/IPV4/IPV6/Ethernet [23].

PTPv2 increases bandwidth efficiency by using shorter synchronization frames, prevents a large number of propagation errors by adding transparent clocks, and provides message extensions by using type-length-value (TLV) parameters. PTPv2 enables different configurations to be setup in specific devices. In PTP these configurations are called “profiles”. The use of profiles helps PTPv2 to be more flexible [15]. Ericsson AB contributed to the development of the first IEEE 1588 Telecom profile to address mobile backhaul needs [23].

2.3.2 **PTP Operation**

PTP provides time synchronization in a network by exchanging messages between nodes based on the IEEE 1588 standard. This standard describes the characteristics of clocks in the system, it determines the current state of the clock, and provides messages to transmit clock information between the network nodes in order to perform synchronization [20].

There can be different types of clocks in a network. For example, a clock could be an atomic clock which is very accurate, a GPS receiver (synchronized to multiple atomic clocks in GPS satellites), or the clock could be some type of crystal oscillator. PTP determines the different types of clocks and creates a master and slave relationship between them. Each node in PTP can be a master, a slave, or even act as a master on one interface and slave on another interface. In a master-slave relation, the master is the more accurate timing device and will be used to synchronize the slave node [21].

2.3.3 **PTP Stack architecture**

PTP protocol can be expressed with two main functionalities: (1) exchanging special PTP messages that provide time information between master and slave and (2) updating the internal clock based on the received time information in order to stay at sync. Figure 2-1 shows the relationship between hardware and software when using IEEE1588. The hardware unit includes a precise real time clock and timestamp unit to generate timestamps, while the software unit implements the PTP protocol. Timestamping is one the most critical actions in synchronization when using PTP. So this timestamping operation should be located as close as possible to the hardware in order to be precise. To achieve nanosecond accuracy in PTPv2 hardware timestamping is necessary [24]. Existing PTP protocol implementation, use various combinations of hardware and software; the details of the implementation affect the accuracy of the system. The following subsections describe some of the different combinations that exist.

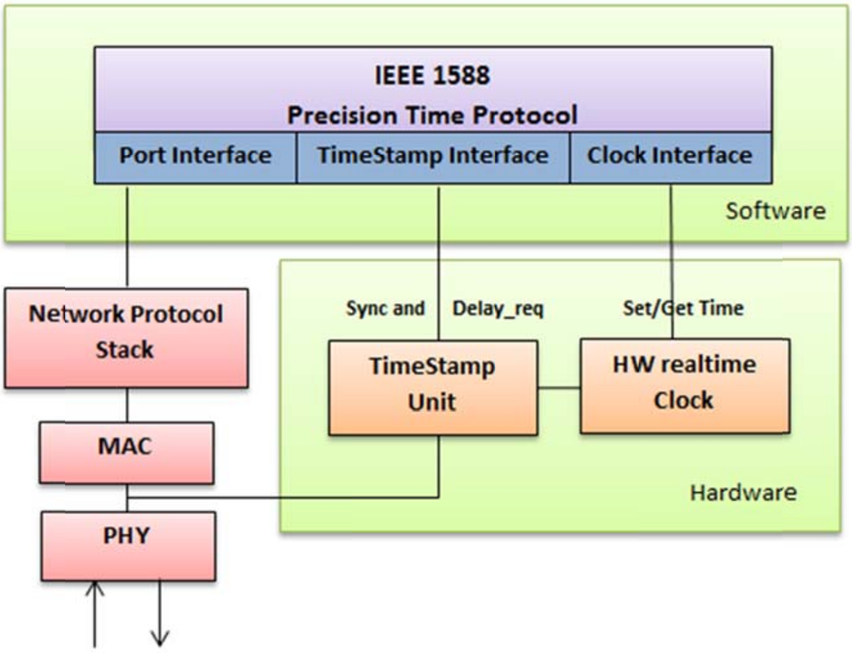

Figure 2-1: General protocol architecture for IEEE1588

2.3.3.1 Software based PTP

In a software implementation of PTP, both the timestamp unit and the IEEE1588 clock are realized in software. Figure 2-2 shows the architecture of the various PTP functions in a software-based PTP system. There are several software-based PTP implementations available in the market, of these two open source implementations will examined in section 3.3 [25].

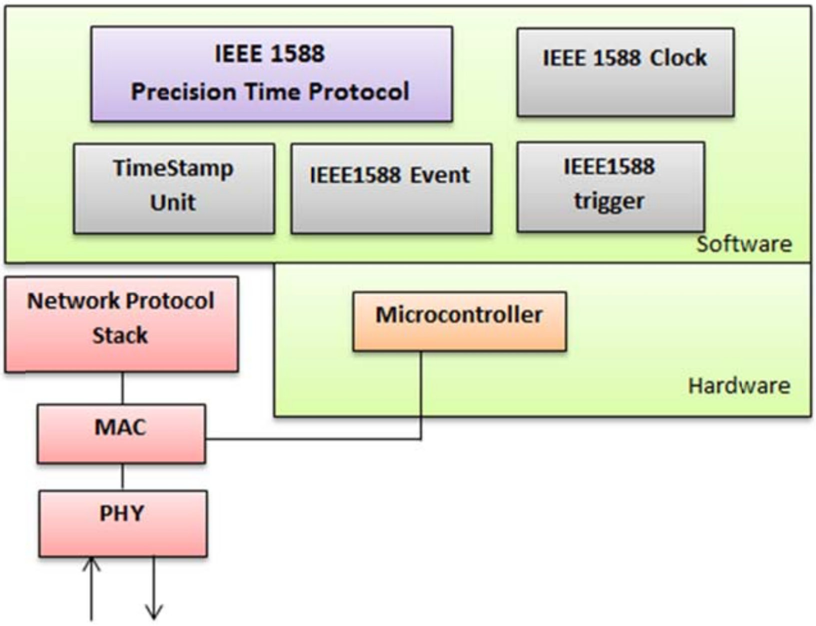

Figure 2-2: Software-based PTP functions

2.3.3.2 Hardware-based IEEE1588

As implementing PTP in hardware enables greater accuracy, vendors have designed commercial PTP implementations in hardware. There several ways of implementing the PTP protocol in hardware, such as Field-Programmable Gate Array (FPGA)-based, chip-based, and network physical layer (PHY) [25]. Each of these will be briefly described in the following paragraphs.

An FPGA-based realization of IEEE1588 is shown in Figure 2-3. The exact design may vary, but in all designs the FPGA is placed outside of communication interface. An FPGA-based design offers greater accuracy than a software-based PTP implementation, as the FPGA directly connects the IEEE 1588 clock and the timestamping subsystem.

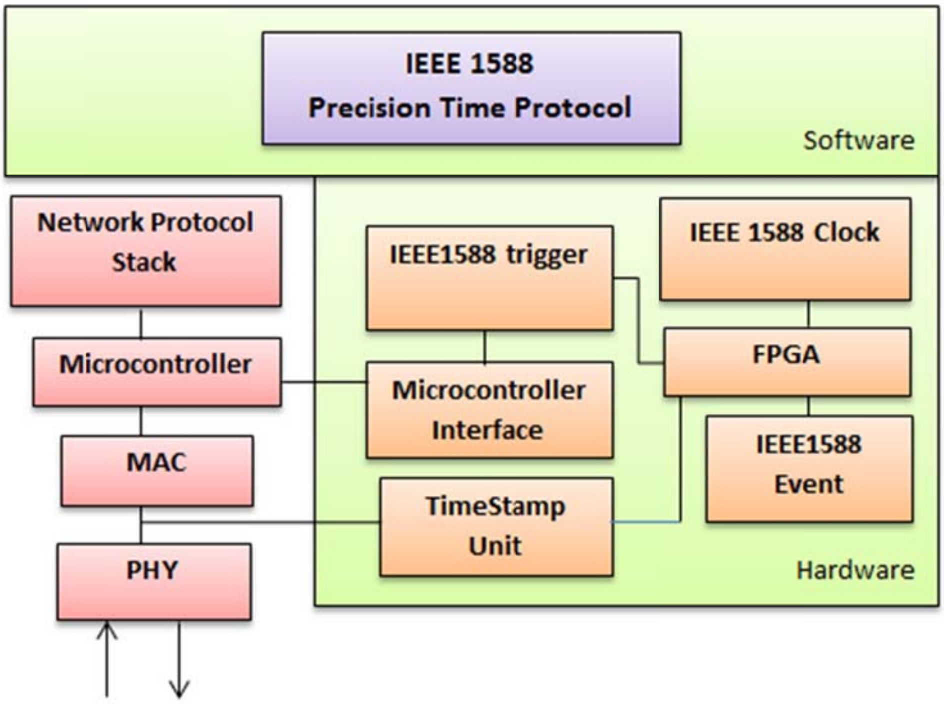

Figure 2-3: **FPGA-based PTP functions**

In a chip-based design, accuracy is improved because the chip includes a dedicated microcontroller and Ethernet Media Access Control (MAC) which performs hardware timestamping when packets enter the MAC. Figure 2-4 shows the general view of microcontroller based PTP system [25].

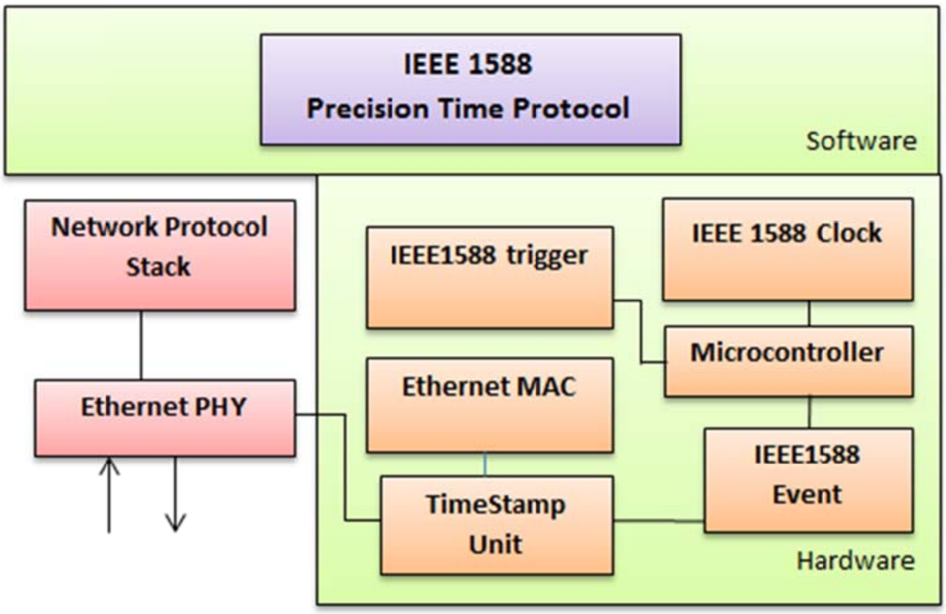

Figure 2-4: **Microcontroller-based PTP functions**

2.3.4 **Components of a PTP network**

There are different types of PTP components that interact in order to achieve synchronization. These components are the grandmaster clock (GM), ordinary clock (OC), boundary clock (BC), and transparent clock (TC). Figure 2-5 shows the PTP master-slave hierarchy in a network. These different types of clocks are described below [26, 27] (based upon [26, 27]):

Grandmaster clocks	The most precise and permanent clock in the system is known as a grandmaster clock. Generally this clock will utilize a GPS receiver or an atomic clock.
Ordinary clocks	A node with a single PTP port that acts as an end device is called an OC. This clock can act as a master or slave in the network according to the Best Master Clock (BMC) algorithm which will be explained in section 2.3.5.
Boundary clocks	A node with multiple PTP ports that acts as a transmission device is called a BC. This clock is usually a standard switch or a router that *also* provides precise timing for the segments that it is connected to. In this clock, one port is chosen by BMC to act as a slave that will be synchronized with a more accurate clock. The other ports of the device will act as masters to synchronize other clocks which will be deemed slaves.
Transparent clocks	TCs were added to PTP in version two. These devices can act as a transmission device. The main task of this type of node is to introduce an interval delay into the system; this improves the protocol with respect to delay compensation and accuracy. In the system, these clocks can act as peer-to-peer transparent clocks or end-to-end transparent clocks. An end-to-end transparent clock introduces an interval delay for end-to-end systems, thus preventing queues from forming in the system. A peer-to-peer transparent clock introduces propagation delay rather than an interval delay, thus it also prevents a queue from forming and helps to provide faster convergence in the case of a change in the network [21, 27].

Figure 2-5 shows an example of the network elements and PTP components needed to exploit the concept of PTP. This figure also helps to visualize the master-slave relationship between these components.

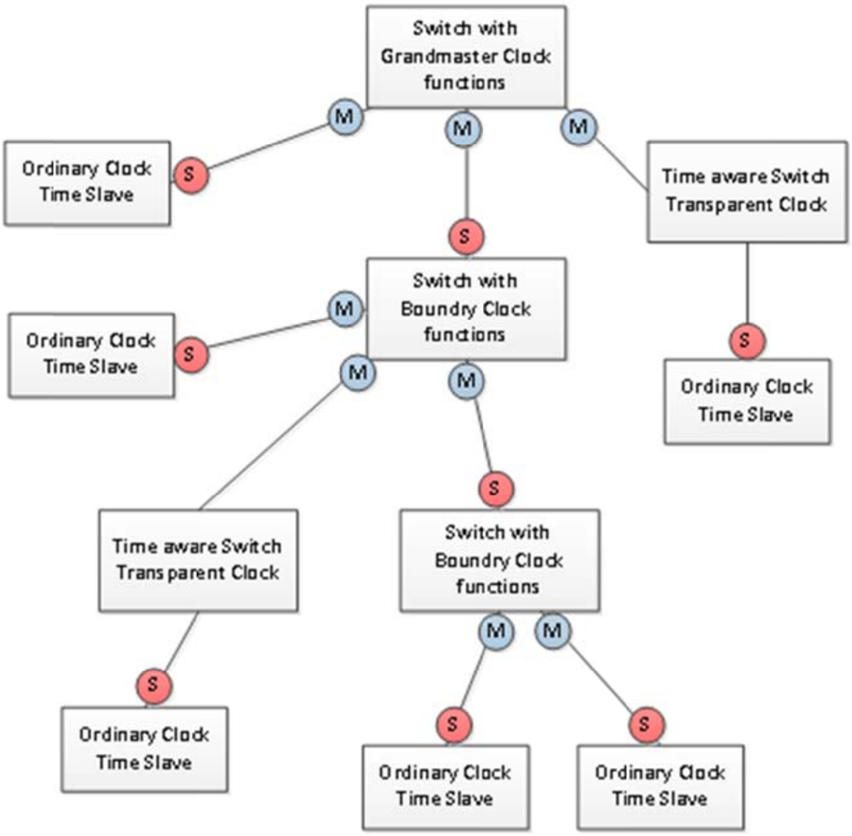

Figure 2-5: **PTP component in network, adapted from [28]**

2.3.5 Best Master Clock Selection

Best Master Clock (BMC) is a distributed algorithm that involves BCs and OCs; it does not involve TCs. The main task of the algorithm is to define master and slave relationships for each communication link. For each communication link the master will be the port of a clock with higher accuracy and the slave is a node that should be synchronized [29].

In PTP, clocks are categorized according to their accuracy by different stratum levels. These accuracies are specified as:

Stratum 1 has an accuracy of within 25 nanoseconds. A stratum 1 clock is considered a primary reference clock. This clock is considered to be a grandmaster clock and is usually either an atomic clock or a GPS receiver [29].

Stratum 2 has an accuracy of within 100 nanoseconds. This clock will be a second level reference in a PTP network [18].

Stratums 3 and 4 can be considered as other references at a lower level of accuracy (greater than 100 nanoseconds) [18].

Stratums 5-254 are reserved [18].

Stratum 255 is a clock with a default setting and can never be considered as a master clock [18].

2.3.6 Messages in PTP

PTP is a synchronization protocol which relies on messages exchanged between master and slave clocks. These messages can be categorized into two types: event messages (which include an origin timestamp) and general messages.

In PTPv1, the event messages are: (1) Sync message and (2) Delay_req message. A Sync message is sent by a master. This message includes best master selection information and an origin clock timestamp. A Delay_req message is sent by a slave clock with the origin timestamp of this slave clock. An origin timestamp is the local node's clock value at the time the message departs the node. A Sync message will be sent periodically to keep the slave's clock up to date.

Follow_up, Delay_resp, and Management messages are general messages. The Follow_up message is sent by a master right after a Sync message and includes a precise origin time stamp. A Delay_resp message is sent by the master in response to a Delay_req and includes the delay receipt timestamp. The communication between master and slave in PTP v1 will be explained in section 2.3.8.

In PTP version 2, Pdelay_req and Announce messages were added to the set of event messages. The Announce message is responsible for best master selection. In PTPv1, this information was included in Sync message. Using a different message for best master selection leads to a smaller Sync message, this can decrease the amount of bandwidth used by PTP in the network. The Sync message will be send more frequently than an Announce message. This reflects the assumption that the network and set of clocks change relatively infrequently and that the primary purpose of the protocol is to provide accurate clock synchronization.

Pdelay_resp, Pdelay_follow_up, and Signalling messages were added to the set of event messages in PTPv2. The Signalling message is used to send additional data using TLV tuples. Pdelay_resp and Pdelay_follow_up messages are used in the new peer delay mechanism [18].

2.3.7 PTP message packet format

Table 2-2 shows summary of message types in PTP and their purpose which discussed in detail in section 2.3.5. Each message contains header, suffix and body. All of the messages have the same header. Header format is shown in Figure 2-6. Table 2-3 shows the specification of each field in the header field of the PTP packets [30].

Table 2-2: PTP messages type and specification

	event messages	**general messages**	**announce message**	**signaling messages**
Message name	Sync Delay_Req	Follow_Up Delay_Resp		
Messages purpose	generate and transport timestamps needed for the synchronisation	used to measure the link delay between two clocks	build up the synchronization hierarchy	all other purposes
UDP port	port 319	port 320		

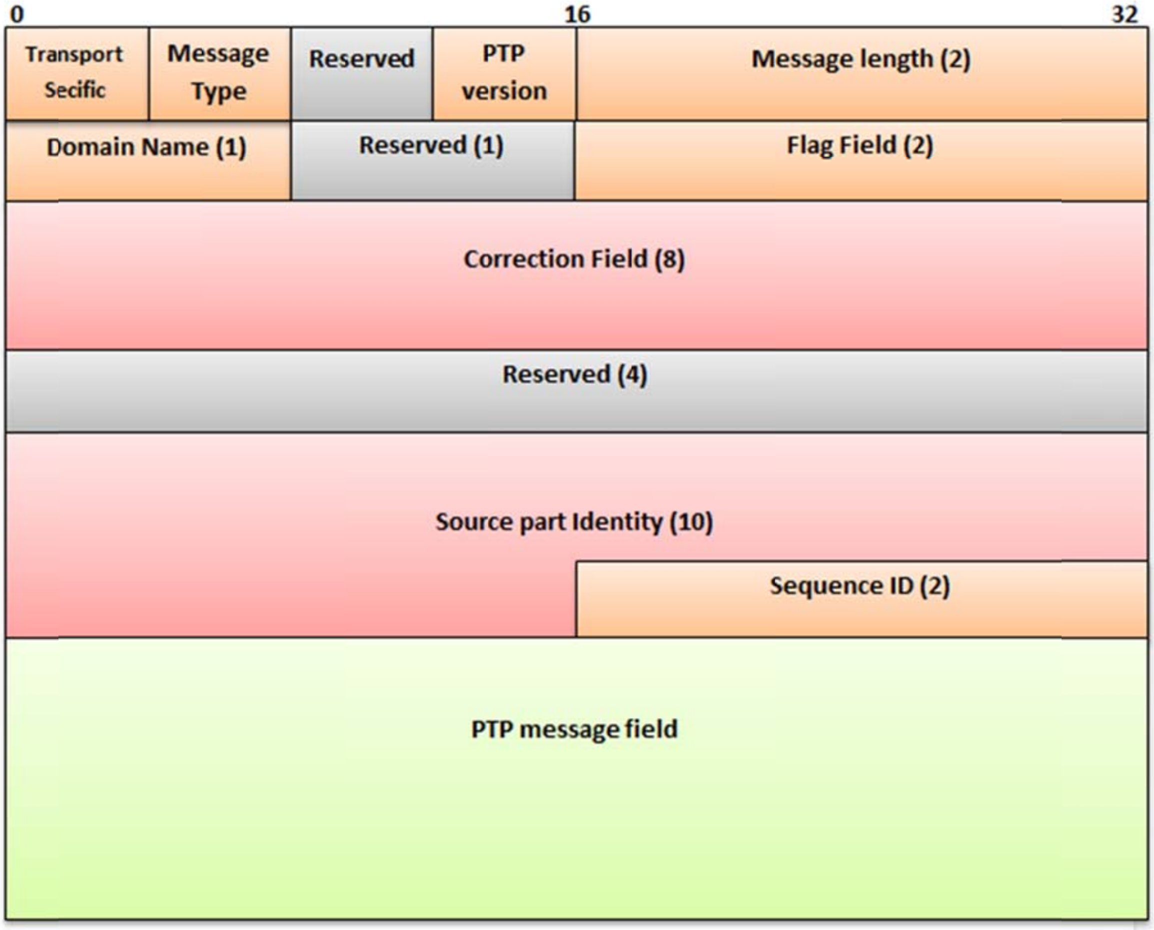

Figure 2-6: PTP header format

Table 2-3: Header fields of the PTP packets [26]

Field	value	Explanation	Octets	Offset
Transport Secific	0	IEEE1588 PTP message	0.5	0
	1	802.1AS PTP message	0.5	0
Message type	0	Sync message	0.5	0
	1	Delay_req message	0.5	0
	2	Pdelay_req message	0.5	0
	3	Pdelay_resp message	0.5	0
	4-7	Reserved	0.5	0
	8	Follow_up message	0.5	0
	9	Delay_Resp message	0.5	0
	A	Pdelay_Resp_Follow_up ,essage	0.5	0
	B	Announce message	0.5	0
	C	Signalling message	0.5	0
	D	Management message	0.5	0
	E-F	Reserved	0.5	0
Version PTP	Version of IEEE1588		0.5	1
Message length	Length of PTP message		2	2
Domain Number	It shows the number of domain the PTP message sender belong to		1	4
Correction Field	This is a correction value which is ns multiplied by 216		1	5
Source Port Identity	Indicates clock ID and port which the message sent		10	16
Sequence Id	Indicates the sequence number of PTP message which relate the messages together		2	30
Control Field	It uses for compatibility to IEEEv1. Different by messages.		1	32
Log Message Interval	Indicates the interval of PTP messages		1	33

2.3.8 **Delay Computation**

In PTPv1, the delay request-response mechanism is used for computing delay. The peer delay mechanism is another delay computation method that was introduced in PTPv2. Figure 2-7 shows how messages are exchanged between the master and slave to compute delay.

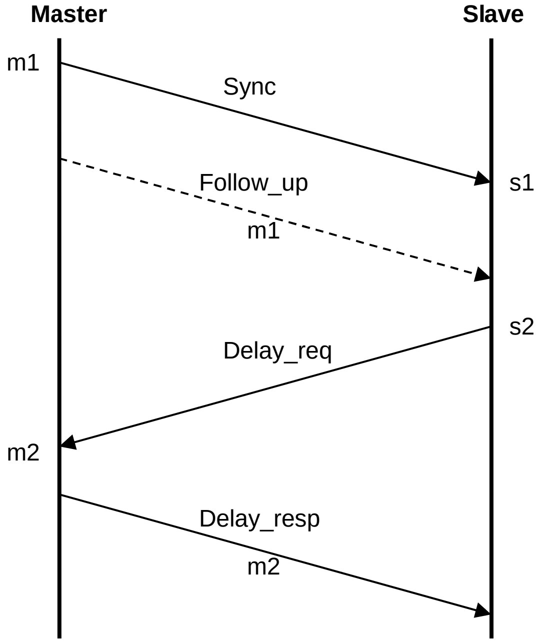

Figure 2-7: Master-Slave message exchange in PTP-v1, adapted from [17]

s1 = slave receives Master Timestamp at this time (via a Sync message)

m1 = master sent the Master Timestamp at this time (included in Follow_up message)

s2 = slave sends a request at this time

m2= master receives the request at this time

t(Master-Slave) = s1-m1

t(slave-Master) = m2-s2

$$\textbf{Delay} = \frac{\text{t(Master - Slave)} + \text{t(slave - Master)}}{2} \qquad \textbf{Equation 2-1}$$

2.3.9 **Delay request-response mechanism**

PTP was designed for symmetric applications; so the delay computation assumes that the sender-receiver delay is equal to receiver-sender delay. In PTP v2, this mechanism can also be used to compute end-to-end delay between transparent clocks [18]. Figure 2-8 shows how messages are exchanged between a master and slave with an intermediate end-to-end transparent clock.

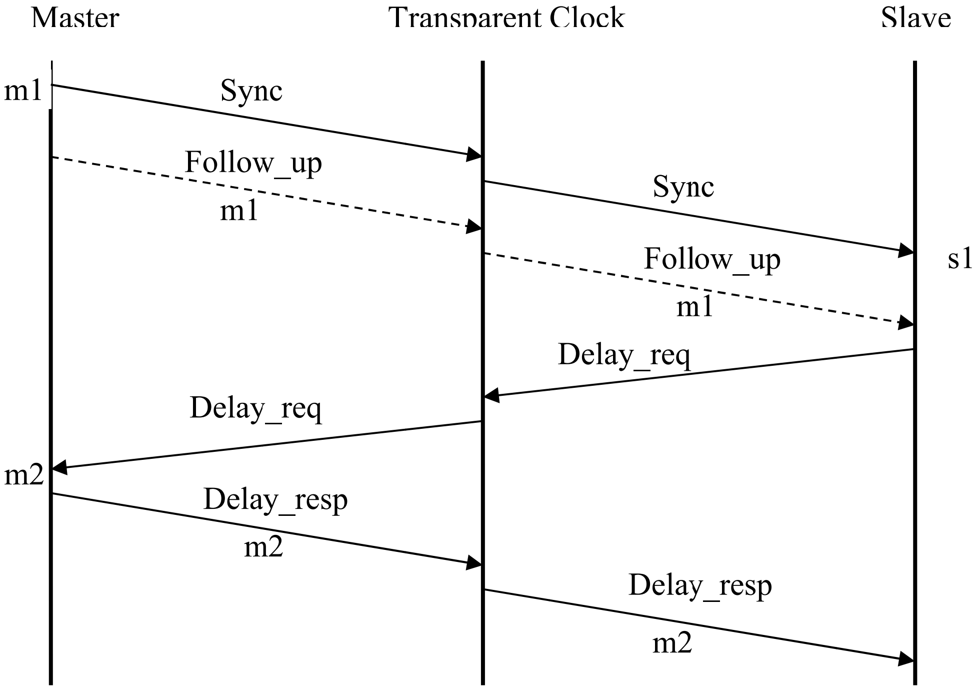

Figure 2-8: **Message exchange with end to end transparent clock, adopted from [16]**

2.3.10 **Peer delay mechanism**

The peer delay mechanism is used when computing the delay between two OCs, BCs, and a peer-to-peer TC. Pdelay_req, Pdelay_resp, and Pdelay_follow_up messages are involved in this mechanism. In contrast to the delay request-response mechanism messages, these messages do not have to be forwarded and are simply used in measuring the point-to-point delay between two nodes [18]. Figure 2-9 shows this mechanism in more detail.

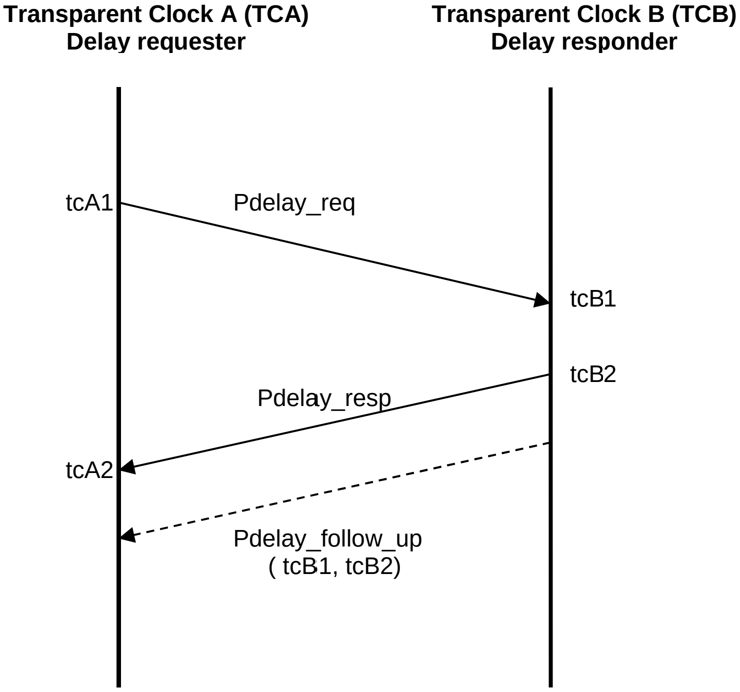

Figure 2-9: **Message exchange with peer to peer transparent clock, adopted from [14]**

tcA1 = the time TCA sends a Pdelay_req

tcB1 = the time TCB receives the request from TCA

tcB2 = the time TCB responds to Pdelay_req

tcA2 = the time TCA receives the response

Transparent clock B (TCB) sends the Pdelay_follow_up after responding to Transparent clock A (TCA)'s response. This Pdelay_follow_up message contains tcB1 and tcB2. So, at the end of this message exchange, TCA is aware of tcA1, tcA2, tcB1, and tcB2. Thus the delay can be calculated as:

t(requester-responder) = tcB1-tcA1

t(responder-requester) = tcA2-tcB1

$$\textbf{Delay} = \frac{\text{t(requester - responder)} + \text{t(responder - requester)}}{2} \qquad \textbf{Equation 2-2}$$

2.3.11 Metropolitan Area Network

A metropolitan area network (MAN) [31] spans a large area of between 5 to 50 Kilometres [24]. A MAN can be used for communication between multiple buildings or even across a city. A MAN is usually made up of several LANs connected via bridges. A MAN is frequently used to connect broadband network subscribers to a larger network such as WAN.

Groups of nodes in a MAN can be classified according to the following hierarchy: access, aggregate, and core. An access network is the part of a MAN closest to the customer. The aggregate part is the middle part that connects the core network to the access network(s).

2.4 Available Synchronization Metrics

In this section the available metrics in network synchronization will be explained. First the two traditional metrics time error (TE) and time interval error (TIE) will be discussed. Then, traditional clock metrics, Maximum Time Interval Error (MTIE) and Time Deviation (TDEV), will be described.

2.4.1 Time Error

TE is a function that calculates the difference in time values of two clocks at a certain time. TE is shown is in equation 2-3[32].

$$\boldsymbol{x(t) = T(t) - T_{ref}(t)} \qquad \textbf{Equation 2-3}$$

2.4.2 Time Interval Error

TIE (shown in equation 2-5) computes the time difference between two clock values at two different times (as shown in equation 2-4 [32]).

$$T(t+\pi)-T(t) \qquad \textbf{Equation 2-4}$$

$$TIE(t,\pi) = [T(t+\pi) - T(t)] - [T_{ref}(t+\pi) - T_{ref}(t)] \qquad \textbf{Equation 2-5}$$

Where:

t is a certain time;
π certain period of interval;

TIE and TE are traditional metrics that can be used to measure the phase error of a local clock to a reference clock (in order to perform phase synchronization). It is also possible to calculate the frequency error using TIE metrics by calculating the slope of a curve of TIE [33].

2.4.3 Maximum Time Interval Error

MTIE is a metric that measures the maximum interval error during a certain period of time. Each period's time interval error can be associated with a sliding window.

$$\boldsymbol{MTIE(n,\tau_0) = max_{1\le k\le N-n}[max_{1\le i\le N-n}\ x_i - min_{1\le i\le N-n}\ x_i]\ ,\quad n = 1,2,..,N-1} \qquad \textbf{Equation 2-6}$$

Where:

n number of sample within a certain period;
τ_0 error sampling interval;
k index of the special period over the data;
N total number of samples;
x_i time error sample;

2.4.4 Time Deviation/ minTDEV

Minimum Time Deviation (minTDEV) is a PTP frequency metric. MinTDE is a mask which is independent of the number of hops or switch time. MinTDE can be used to evaluate whether the network meet a synchronization requirement or not [10].

$$TDEV(n\tau_0) = \sqrt{\frac{1}{6n^2(N-3n+1)} \sum_{j=1}^{N-3n+1} \left[\sum_{i=j}^{n+j-1} (x_{i+2n} - 2x_{i+n} + x_i) \right]^2}$$

$$n = 1, 2, \ldots integer\left(\frac{N}{3}\right)$$

Equation 2-7

2.4.5 Modified Allan Deviation (MDEV)

MDEV is a traditional algorithm designed to increase the precision of periodic signal frequency by reducing the total observation factor (division by n), and also by diving the time interval τ into n cycle clock period [34].

$$\mathbf{MDEV}\ (n\tau_0) = \frac{\sqrt{3} * TDEV(n\tau_0)}{n\tau_0}$$

Equation 2-8

2.5 Related work

After PTPv2, the IEEE 802.1AS “Standard for Local and Metropolitan Area Networks - Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks” standard based on the second version of the PTP was published. Although this standard has the string “for LAN and metropolitan area networks” in its name, this protocol is actually for time sensitive application, such as audio and video. This standard does **not** explain how PTP will operate in larger networks. This standard is explained in section 2.5.1 in more detail.

Some experiments have been done regarding the performance of PTP in larger networks; these are explained in more detail in section 2.5.2.

2.5.1 IEEE 802.1AS

IEEE 802.1AS is a timing and synchronization standard for the transport of time sensitive applications such as audio/video in local and metropolitan area networks. This standard is based on IEEE 1588 v2 and includes a PTP profile[35]. A summary of the different mechanisms of this PTP profile are described below:

Management mechanism	This PTP profile uses a Simple Network Management Protocol Management information base (SNMP MIB) as its management mechanism.
Path delay mechanism	A peer delay mechanism will be implemented through a logical point-to-point link and it operates separately and independently in each direction of the link.
	A BC is synchronised to the grandmaster though its slave port and synchronizes OC through its master port.

	Usually links are considered symmetric in this profile and offsets will be the mean of the master-to-slave propagation time and the slave-to-master propagation time.
Default Interval and Timeout Values	A Sync message will be send every $\frac{1}{8}$ second. An Announce message will be send every second. A Pdelay message will be sent every second. An announce receipt will time out following two announce message intervals. A Sync receipt will time out following three sync message intervals.
Transport mechanism	Initially PTP was designed for full duplex IEEE 802.3 networks, but PTPv2 supports the different IEEE 802 transport mechanism.
Node types	Each node in IEEE 802.1AS is considered as a time aware system. There are two kinds of nodes in this profile: a middle bridge which acts as boundary clock and end station which acts as an OC.
Optional features	Each end station should accumulate the frequency offset to the nearest neighbour and the frequency offset is carried in TLVs and used to correct propagation and synchronised time.

2.5.2 IEEE 1588v2 over Wide Area Networks

In [36], Novick, et al. studied different PTP hardware in unicast mode. They investigated the performance of this equipment across a WAN. They did their experiment by check the timing output of master clocks which were locked to a built-in GPS receiver in five scenarios. They compared PTP client performance on the same subnet, with a master outside of the local networks, with a remote master connected via a leased (T1) line, with remote master on public Internet, and with a long-distance master connected via a leased (T3) line[*].

This study considered several wide area configurations, such as Internet and virtual LANs. They used four different PTP clients in their experiment.

In their experiments, the different PTP hardware showed different behaviors. Moreover, with increasing distance, the number of devices along the network path will likely increase causing increased *asymmetry* of the path and causing longer delays [36].

2.5.3 Synchronization of GSM and UMTS basestations with PTP

Symmetricom studied the challenges that need to be met when deploying the IEEE1588 standard in a packet switched network in order to provide frequency synchronization for GSM and UMTS-FDD basestations. They considered synchronization requirement for RBSs and also a packet network's timing performance by allocating a performance budget to different element of the network. They use the pyramid noise budget allocation which allocates 10% to the GM, 40% to packet network, and 50% to end equipment (base station) [10].

---

[*] A T1 line is a TDM link operating at 1.544 Mbps and a T3 is operating at 28 times this speed, i.e., 44.736 Mbps.

2.5.3.1 Network Topology

Symmetricom suggests a maximum of four networks hop in a PTP span based upon their empirical study by considering the packet delay variation and performance of a particular device. They also suggest 80% of network load as the maximum traffic load in the network (i.e., leaving 20% of the network capacity available for PTP and traffic bursts). A larger number of hops will require a more stable oscillator or a different traffic load (and traffic pattern).

2.5.3.2 GrandMaster (GM)

Symmetricom allocated 10% of the performance budget to the GM because some inaccuracy may cause because of quantization error. This problem can be eliminated by locking the GM to a primary clock reference. Placement of GM in the network is critical because it will determine the overall accuracy. Symmetricom suggests some factors that should be considered by network planner regarding the placement of GM: potential network load, congestion, and performance of networks elements. In order to avoid accumulation of delay or network jitter, GM servers should be place as close as possible to the edge devices (i.e., the PTP slaves).

2.5.3.3 Metric

Symmetricom developed a metric, minTDEV (described in section 2.4.4) as a proper metric to determine whether the PTP client can be met the frequency accuracy requirement or not.

2.5.4 Other related work

While the focus of this thesis is on extending the IEEE 1588 protocol to MANs, Waheed Ahmed [37] has proposed a solution to increase the accuracy of synchronization in networks by combining the IEEE 1588 synchronization protocol and the Adaptive Oscillator Correction Method (AOCM). He used AOCM to train a clock locked to GPS and applied this method to the master and slave in order to improve stability in handover mode. He implemented a clock agent and a basic IEEE 1588 message exchange in NS-2 that calculates the offset between master and slave implementation

Jens Steinhauser [38] analysed the performance of the PTP protocol and clock synchronization in this protocol in different situations by implementing a PTP like protocol with the OMNET++ simulation tool. Instead of integrating an existing PTP stack, he designed a minimal PTP protocol that does not support the BMC algorithm.

Y. Liu and C. Yang [39] designed an OC in PTP with OMNET++ to study its behaviour. When designing this OC, they used a clock servo* time stamping mechanism and they considered a typical end-to-end PTP process for synchronization. Unfortunately, their simulation package is not a complete package to evaluate the performance of the PTP protocol in different networks as there is still a need to design the other PTP devices, such as BC and TC.

---

* Clock servo is a self-tuning clock that provides correction and clock recovery.

G. Giorgi and C. Narduzzi [40] studied the behaviour of PTP synchronization by implementing a PTP protocol stack in the OMNET++ simulator. In their study, they considered sources of uncertainty, such as timestamp noise and glitches, and also cross traffic according to a given load profile. Similar to Liu and Yang, their node simulation is based on a clock servo design which results in solutions that can be used to evaluate this design in a controlled environment. Their simulation model consists of a master, a slave, a traffic generator, an end device, and a switch at the highest level.

R. Chen, Y. Zhang, C. Cao, Y. Zhao, B. Li, J. Zhang, and W. Gu [41] focused mainly on the effects of different network characteristics of Transport Multiprotocol Label Switching (T-MPLS) on the PTP protocol. They considered queuing disciplines and link capacity. In particular, they consider the effects of two queuing disciplines (First In First Out (FIFO) and priority queuing (PQ)) on the network. Their experiment simulated a network with one PTP master and four PTP slaves using OPNET. Their simulation shows that asymmetric traffic can cause a time offset between master and slave. In PQ in which PTP packets are prioritized, the time offset is a few micro seconds, while FIFO results in time offsets of hundreds of microseconds.

Pedro V. Estrel and Lodewijk Bonebakker[42] discuss the main challenges in providing microsecond accuracy between two servers in a worldwide company using PTPv2. The main issues were characterized into three main areas: definition weakness in PTP protocol, PTP expansion problem, and impact of migrating from a current test bed to PTP.

To monitor the stability of synchronization, they suggest defining the confidence interval of synchronization, but there is no field in the management message to use for this purpose. While finding a suitable single multicast address is challenging when deploying PTP over a WAN, they decided to use multicast for all public PTP packets and unicast for end to end packets. They used several GMs in their topology to cover the whole network and they found that finding the BMC was challenging as several MCs were in different countries. They used several multicast distributions and utilized Protocol Independent Multicast-Sparse Mode (PIM-SM), so each client connects to the nearest Rendezvous Point (RP).

3 Method

This chapter provided the information required for the design and implementation of a solution to answer the question which motivated this thesis problem (i.e., can PTP be used in MAN for RBS synchronization) and to achieve the goals stated in section 1.2. The chapter begins with a description of several possible solutions and gives the reasons why each of them was not used.

As this thesis deals with synchronization in a MAN, it worth mentioning the main characteristics of a packet switched network [39], specifically in a switched network, these characteristics are: switches, network topology, and traffic pattern. The characteristics of an Ethernet switch will be discussed then based upon these characteristics and the characteristics of a packet switched network a solution for solving the scientific question of this thesis will be proposed.

3.1 Network Configuration

The following subsections describe the switches that will be used in the network, the network's topology, and the traffic load that will be assumed.

3.1.1 Switch

A switch is the main device used to interconnect network devices. In this thesis we will focus on the delay characteristics of Ethernet switches. There are four steps in switching a packet within a switch that can cause delay: Input, Output, Scheduling algorithm, and queuing discipline.

Input delay	Classification and traffic management are two operations that may occur when a frame arrives at the switch. Classification concerns identifying the flow to which each frame belongs. Traffic management refers to operations such as admission control, marking, shaping, and policing the traffic. The time it takes for a frame to be processed depends on switch's design and incoming load of traffic to the switch.
Scheduling algorithm	Different switches utilize different scheduling algorithm when they switch frames from the inputs to the outputs of the switch. This algorithm causes part of the delay for packets passing through switches.
Queuing discipline	Different queuing policies effects the output delay before a frame can be output from the switch. As it mentioned in section 2.5.4, R. Chen, et al. [32] studied the effect of FIFO and PQ disciplines on PTP synchronization.
Output delay	Output delay is a delay that added by the priority of traffic and the queuing policy. An example of this kind of delay is "head-of-line blocking" delay, where transmission of the packets that are at the front of the output queue delay packets behind them in the queue.

3.1.2 Network Topology

Networks can have different kinds of topology, such as bus, three, ring, mesh, or some combination of these topologies. The most important impact on the delay is the number of nodes along a path, while the topology itself is generally not a concern.

According to ITU-T G.8261 [43] an interconnected ring topology is usually deployed in a MAN and each such ring contains 10 switches, which gives the maximum distance of 5 switch [28]. For this reason in our experimental study, different numbers of switches up to a maximum of 10 switches were used to investigate the effects of increasing the number of nodes along a path on PTP functionality.

3.1.3 Traffic Load

Traffic in the networks can be bursty or constant. In reality, traffic is bursty in most networks.

According to the ITU-T G.8261 recommendation [43], core networks mostly carry data and minor conversational traffic. The traffic mix is 60% 1518 octets, 30% 64 octets, and 10% 576 octets sized packets [39]. In contrast, access networks mostly carry conversational traffic and streaming, interactive, and background data. The access network packet length distribution is 80% 64 octets, 15% 1518 octets, and 5% 576 octets [32].

In order to examine the effects of different kinds of traffic, both bursty and constant traffic are considered our experimental study. Moreover, the effects of different packet sizes on the network are examined by testing with the minimum and maximum size network frames.

3.2 Simulation

The initial plan to answer the thesis question was to use a simulation of the PTP protocol in order to analyse PTP in a number of scenarios in a large network. The potential alternatives for simulating PTP in larger networks are described in the following subsections.

3.2.1 Network Simulator (NS)

NS is an open source discrete-event network simulator mainly developed for simulating internetworking systems. It introduced in 3 versions: NS-1, NS-2 and NS-3. NS-1 supports some internetworking functionality, such as TCP, dynamic routing, and multicast routing [33]. NS-3 is the latest version of NS that supports research on both IP and non-IP networks.

During the design phase of this project, I looked for an implemented of the PTP protocol in NS. Waheed Ahmed [37] implemented a clock agent and a basic IEEE 1588 message exchange in NS-2 that calculates the offset between master and slave implementation. More detail about his studies explained in section 2.5.4.

Kaiyu Hang [45] researched the effects of various network characteristics on the quality of service (QoS) in mobile networks. He designed a traffic management system implemented with NS-3 implements buffers, queues, schedulers, a traffic shaper, and classifiers. Using an NS based router with support for the IEEE 1588 protocol was a good means to design and evaluate different scenarios to the behaviour of the PTP protocol. His traffic management system mainly works with a NS-Click[*] based node, but an NS-Click converter is used for situations such as PPP encapsulation which is not supported by a Click based node. Figure 3-1 shows his traffic management system.

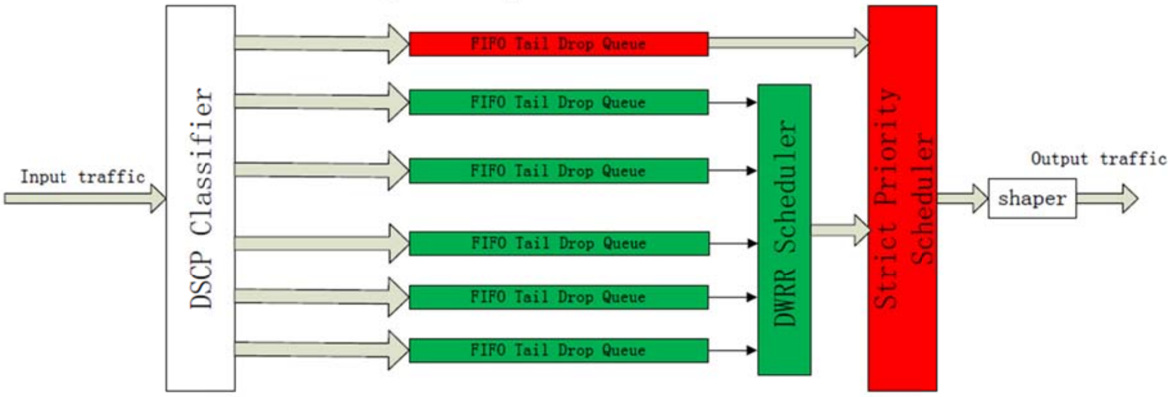

Figure 3-1: **Traffic Management System (adapted from [34])**

3.2.2 OMNET++

OMNET++ is an open source discrete event based simulation environment that was mainly designed for simulation of communication networks. This modular simulation tool consists of extensible components written in C++. These components are assembled into a larger module which supports a graphical user interface and runs on several different platforms.

Y. Liu and C. Yang [39] and G. Giorgi and C. Narduzzi [40] implemented PTP protocol stack in OMNET++. More explanation of their work explained in section 2.5.4.

3.2.3 OPNET

Optimized Network Engineering Tools (OPNET) is a commercial networking simulation package with lots of features and tools that can help a programmer to design protocols and different layers of a network stack. OPNET Modeler is the first OPNET product written in C. This package is mainly oriented for researchers and for use in industry. OPNET Modeler is an event driven network simulator tool that is divided into three levels of modelling:

Network Model	In this model, network objects (subnets and links) and node objects (modules) will be created. Object attributes will be defined and attribute values will be assigned.
Process Model	This model contains objects (states) and state connections (transitions). Examples of this type of model are: communication protocols and algorithms, shared resource managers, and queuing disciplines.
Node Model	A node model consist of a node editor that defines a packet stream, statistic wire[†][47], logical associations, transmitters, receivers, and antenna for the simulation of wireless technologies [37].

---

[*] Click Router is a modular router that uses new software architecture. It is assembled from several individual elements which provide simple router functionality, such as classification, queuing, etc. [42]

[†] In the node domain, a statistical wire refer to interfaces between blocks of queues, processors, and receivers that carry a single data value from source to destination

3.3 Emulation

Another way to use the PTP protocol is to use PTP software stacks. At the time this thesis was written several PTP software implementations are available, such as Oregano Systems' syn1588 PTP Stack, PTPd, the Linux PTP Project, etc. Among these PTPd and the Linux PTP Project are open source, hence they have been studied further in this thesis. Both of these implementations use a clock servo design to provide a software clock and also to support the IEEE 802.1AS protocol.

PTPdaemon is a portable implementation of a standard clock following IEEE 1588v1 and v2 [37]. It supports time and frequency synchronization and is mainly design for testing and measurement. PTPd is a software based system designed as two parts: a protocol stack and a clock servo. Tamás Kovácsházy and Bálint Ferencz [38] have tested PTPd in different configurations. According to their tests, 1 μs accuracy is possible when the network is fully IEEE 1588 aware.

There are two restrictions that should be considered in any system using PTPd. First; it uses a software timestamp to record the sending and receiving time of the message as implement in a software layer. Second, it uses a software clock using a time value that is stored in memory [39]. The PTP protocol timing accuracy is based upon the accuracy of the time stamp and the accuracy of the oscillator of the PTP clock. A software timestamp is not as good as a hardware timestamp and a software clock using the computer's clock does not have an accurate oscillator.

The Linux PTP project is another software implementation of IEEE 1588 based on Linux. It supports both hardware and software timestamps via the Linux SO_TIMESTAMPING socket option. Moreover, it supports a Linux PTP Hardware Clock (PHC) subsystem, leap second handling, and path trace TLV. Table 3-1 gives an overview of the features of both PTPd and the Linux PTP project [40].

Table 3-1: PTPd vs Linux PTP project

	Runs on	Tiemstamping	Clock	Transport
PTPd	Linux, uClinux, FreeBSD, NetBSD	Software timestamp	OC	Raw Ethernet UDP/IPv4, UDP/IPv6
PTP Linux Project	Linux	Hardware(PHC) or software timestamp	OC/BC	Raw Ethernet UDP/IPv4, UDP/IPv6

In order to do emulate a time aware network, we need an OC and a BC. As explained earlier, a BC is a node with more than one port that intercepts incoming PTP messages and forwards them for management purposes. The following are some of the important points that should be considered in order to emulate a BC:

1- Having more than one port,

2- The interwork interface card should support hardware timestamps, as a software timestamp is not accurate enough, and

3- A node with several independent network interfaces is not sufficient, as all the ports should share a single clock.

These are some test and measurement boards that provide several ports with a shared clock that are suitable for emulating a BC. For example, the FreeScale P2020E/P2010E Reference Design Board is a suitable board for emulation of a BC [49]. Figure 3-2 shows an example architecture for emulating a BC using open source software and such a test board:

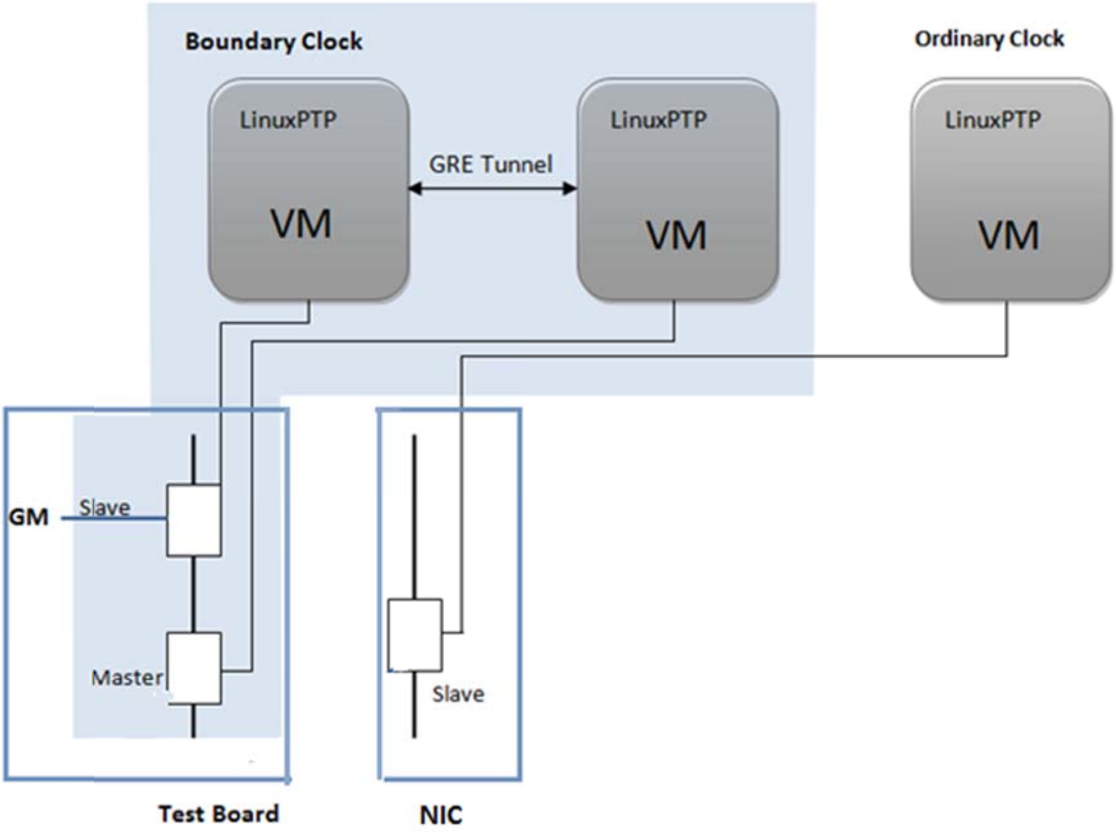

Figure 3-2: **BC emulation design**

3.4 Experimental Study

This section describes our experimental study of the IEEE 1588 standard. During this experiment, the delay between a PTP GM and PTP slave were measured. In the first part of this section the goals of the experiment are explained. Following this the equipment used for these experiment and each scenario will be described.

3.4.1 Goals

As mentioned in chapter one, the PTP protocol was mainly designed for small networks such as LANs. The main goal of this thesis project is to investigate if it is possible use the PTP protocol in a large packet switched network. In this investigation several different factors that might affect PTP performance are considered.

The goal of the experiments is to measure the *relative* delay between a PTP GM and a PTP slave with different numbers of nodes along the path and under various traffic loads. After these measurements the data will analysed to see if the synchronization requirements of the different RBS technologies can be met.

3.4.2 Environment Setup

In this study, more than 150 tests were performed over a period of 10 weeks. In these tests, one PTP GM and one PTP slave were used together with different numbers of switches between them. For business reasons the switch vendors will be referred to as A, B, and C in this document. The specification of the switches is explained in appendix A where the vendors are listed in alphabetical order. The physical link between switches and PTP nodes in the network is CAT 5e cable with a length of 0.5 to 2 meters. The two PTP aware nodes were connected to an oscilloscope (details about the oscilloscope will be presented in section 4.2.2) to measure the time difference (and drift[*]) of the GM and slave with different numbers of switches between them and when the network has different traffic loads.

In this study virtual bridge is used to emulate the network characteristics and study the PTP switch behavior in those situations (delay and packet loss), the detail of this experiment is explained in 3.4.4 and 0. Table 3-2 shows the hardware specification of a computer that used to run the virtual bridge package. Table 3-3 shows the specification of Network Interface Card that used to connect GM and slave together [54].

Table 3-2: Hardware Specification

	Specification
Central Processing Unit (CPU)	Intel(R) Core(TM)2 Duo, 2.3 GHz
Memory	4 GB
Operating System (OS)	Linux Ubuntu 12.4

Table 3-3: Network Interface Card (NIC) Specification

	Specification
Model	D-Link DFE-580TX
Board	DL10050
Cache buffer	2K byte receive FIFO 2K byte transmit FIFO
Standard	IEEE 802.3 10BASE-T Ethernet IEEE 802.3u 100BASE-TX Fast Ethernet PCI local bus 2.2 specifications IEEE 802.3x Flow Control IEEE 802.1P priority tagging PCI 2.2

---

[*] The drift that we measure is the change in time difference between the two nodes over a period of time.

3.4.3 Scenarios

In this section a general overview of the scenarios will be discussed. Figure 3-3 shows the network topology used for the various scenarios.

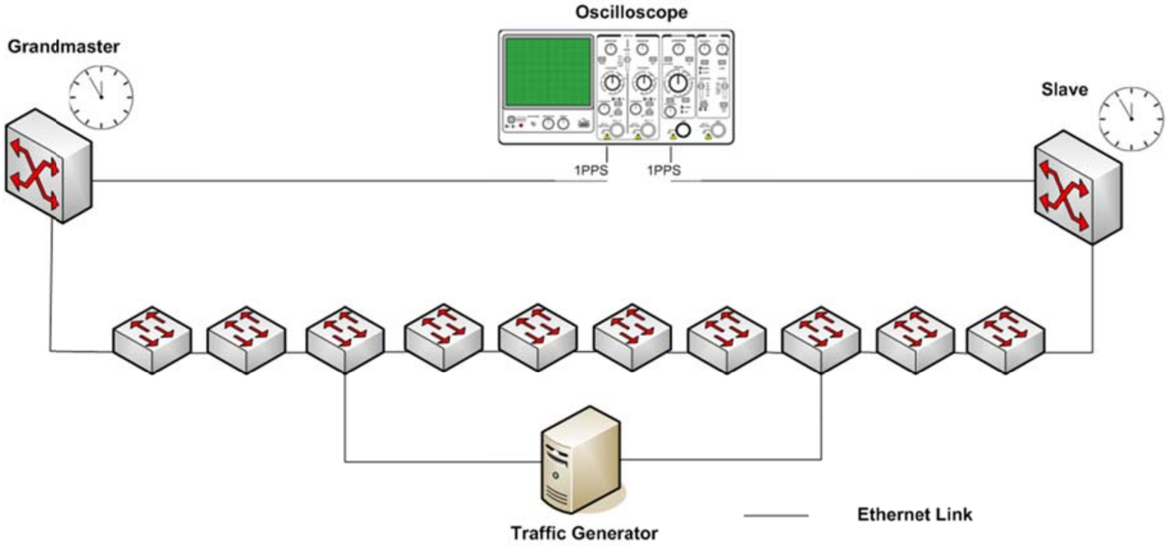

Figure 3-3: Overview of network scenario in the experimental study

Table 3-4 shows a summary of the scenarios that will be discussed in section 4.3. In this thesis nine different scenarios were tested to evaluate the delay of the network with respect to two independent variables:

Number of node	In these scenarios the relative delay between GM and slave are measured. Initially no node is on the path and the number of nodes on the path is incrementally increased to 10 nodes between the GM and the slave.
Traffic Load	Different kinds of traffic pattern are injected into the network. Table 3-5 shows the traffic profiles that were used in the experimental study.

Table 3-4: Summary of Scenarios and Tests (with × indicating a combination of number of nodes and load that was tested)

Load \ Number of Nodes	0	1	2	3	5	6	7	9	10
Profile 0	×	×	×	×	×	×	×	×	×
Profile 1	×	×	×	×	×	×	×		×
Profile 2	×	×	×	×	×	×	×		×
Profile 3	×	×	×	×	×	×	×		×
Profile 3	×	×	×	×	×	×	×		×
Profile 4	×	×	×	×	×	×	×		×
Profile 5	×	×	×	×	×	×	×		×
Profile 6	×								
Profile 7	×	×	×	×	×	×	×		×
Profile 8	×	×	×	×	×	×	×		×

Table 3-5 presents the profiles used in the experimental study of this thesis project. Each profile defines a traffic load specification. Profile 1 to profile 4 considers different constant traffic load specification. On the other hand, profile 5 to profile 7 defines burst traffic load specification. To be more specific, profile 5 and profile 6 use 64B frame packets. Profile 7 uses 1518B packets. Profile 8 indicates full bandwidth traffic load, which is 1518B traffic.

Table 3-5: Traffic pattern and traffic load specification

	Profile							
Specification	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Traffic Type	Constant				Burst			Constant
Traffic Mode	Continuous							
Load %	25%	50%	75%	100%	50%	80%	80%	100%
Load Frames	372023.8	744047.6	1116071.4	1488095.2	744047.6	1190476.1	1190476.1	1488095.2
Load L2 Mb/s	190.476	380.952	571.428	761.904	380.952	609.523	609.523	761.904
Load L3 Mb/s	136.904	273.809	410.714	547.619	273.809	438.095	438.095	547.619
Frame Integrated Device Technology IDT (ns)	2688	1344	896	672	672	672	672	672
Burst Integrated Device Technology IDT (ns)					672672	16867272	16867272	
Burst Length Frame					1000	1000	1000	
Burst Load Percentage					100%	100%	100%	
Burst Load Frames					1488095.2	1488095.2	1488095.2	
Burst Load L2 Mb/s					761.904	761.904	761.904	
Burst Load L3 Mb/s					547.619	547.619	547.619	
Packet Layer	Layer 2 Frame							
Packer Length (Byte)	64						**1518**	

3.4.4 Network Characteristics Emulation Scenarios

The internal algorithm used by the PTP slave that used in this thesis project is not known; hence we do not know how quickly the slave will discipline its local oscillator. In order to estimate the convergence time of the switch in different situations such as packet loss and delay, we emulate the network characteristics in a controlled way to monitor the behaviour of our PTP switches in such system.

Our solution to find out the effect of delay and PTP packet loss in the network on PTP switch used in this thesis project is to connect GM and slave through a virtual bridge together, then use Traffic Control (TC) to emulate network characteristics. Two scenarios is defined in this thesis project to estimate expected result of our PTP switches. One is mainly focused on adding fixed amount of delay which can be associated to the fixed delay added by network devices such as middle switches in networks. The other scenario mainly focused on effect of PTP packet loss on our PTP switches which is associated with bursty traffic in the network that cause packet loss. Detail of each scenario is discussed in section 0.

Figure 3-4 shows the network topology used for two scenarios in which the network characteristics were emulated.

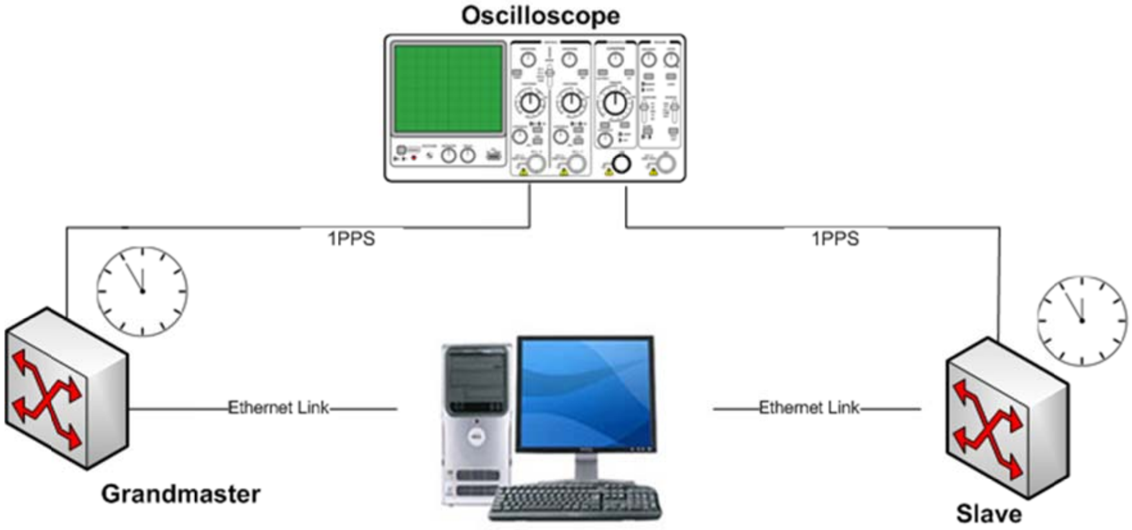

Figure 3-4: GM andslave connected via virtual bridge

4 Design and Implementation

This chapter focuses on design and implementation of the solution proposed in this thesis. In section 4.1 the OPNET simulation will be discussed. In section 4.2 the details of the equipment and a specification of the testbed used in this thesis will be given. The details of each of the test scenarios will be discussed in section 4.3. In last section of this chapter, the data collection process will be discussed.

4.1 Simulation with OPNET

As it mentioned in section 3.2.3, OPNET is a powerful simulation tool that provides a user with the ability to design a network protocol and simulate different test cases. During the initial work in this thesis project, a simulation of PTP protocol was attempted in order to compare these simulation results with commercial PTP hardware.

The PTP simulation designed by Depari, et al. was chosen for this project. This simulation was developed as a network model by adding two nodes: OC and BC [55]. In the node level, the default OPNET network components were used together with the addition of new UDP/IP modules in the OPNET Ethernet stack model to provide a PTP clock model. At the process level a PTP engine was designed as a state machine that determines the protocol's action in different situations.

The PTP engine in their design consist of 4 main states: PTP master, PTP slave, PTP passive, and PTP pre-master for handling the protocol, together with 3 states for error handling. This thesis project attempted to improvement this PTP engine in order to handle BCs and TCs by adding two states: PTP BC and PTP TC. Although their PTP engine supports a BC, it was mainly design for handling several GM and found the best master by using the BMC algorithm. The proposed PTP BC state would be responsible for performing time correction when the BC receives PTP packets. Figure 4-1 shows the suggested state processing module for PTP engine.

Another suggested modification is to improve the PTP functionality. In the node level, the TC should be added by using a default a switch and modifying the node attribute and also designing three additional packets that are used in the peer delay mechanism in TC. As was mentioned in section 2.3.6, PTP headers are the same for all PTP messages, but the suffix and payload are different.

However, as was described in section 1.3 the effort to modify their simulation code was terminated due to the lack of some of their source code and my inability to replace this missing code.

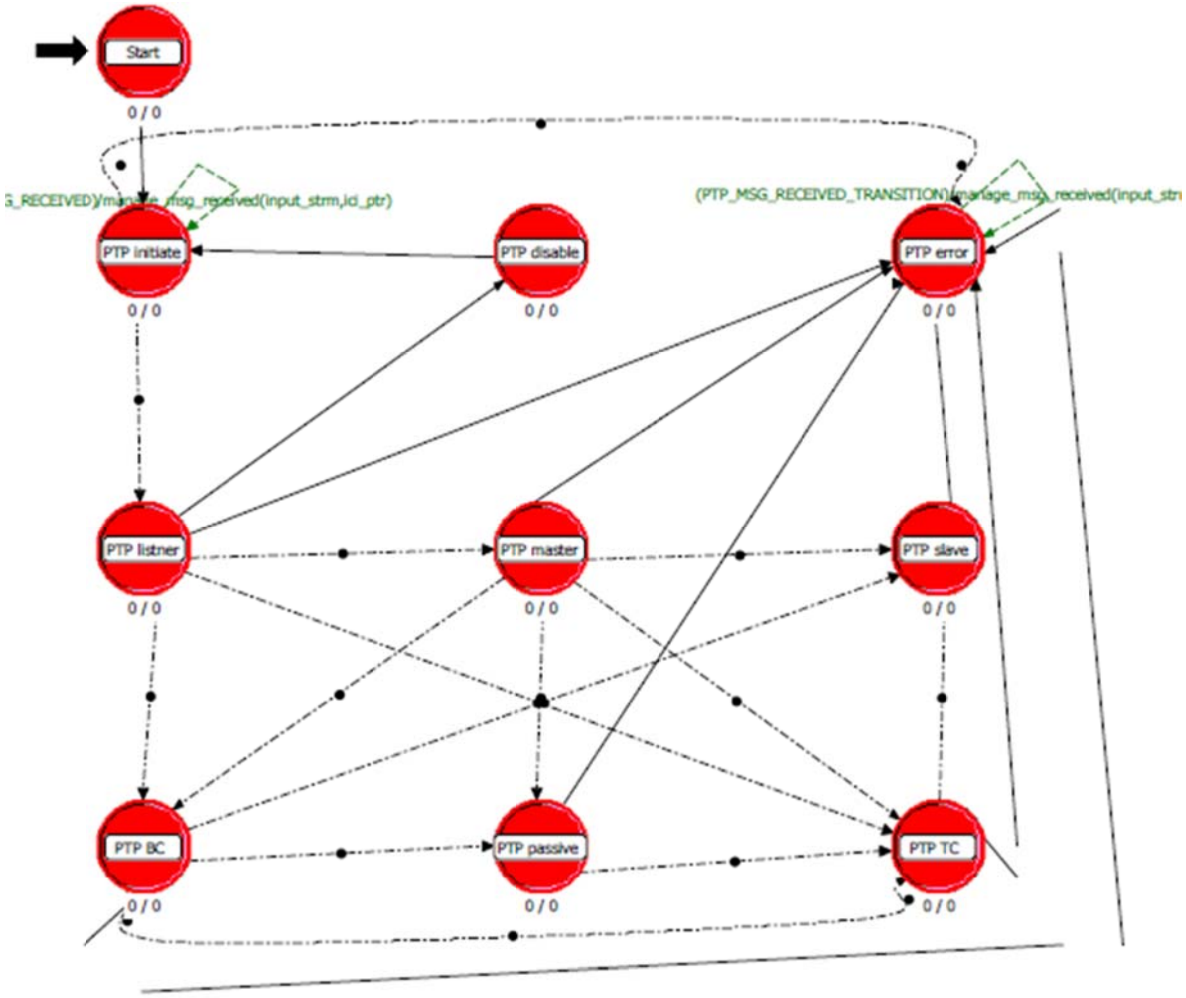

Figure 4-1: **Suggested new PTP engine process**

4.2 Testbed

This section describes the PTP nodes and the oscilloscope that were used for the experiments. As was stated in section 3.4.2 the details of the specific switches that were used are purposely left out of this description.

4.2.1 PTP node Specification

The time aware devices that used in this thesis experiment used the FPGA design for a PTP implementation as described in section 2.3.3.2. The specifications for each node are given in Table 4-1. In this table PRC refers to a Primary Reference Clock, i.e., a Stratum 1 time source.

Table 4-1: PTP node specification

	Protocol Version	Best accuracy	PTP implementation	PTP Messages Interval
Grand Master	PTPv1	100 µs without PRC 10ns with PRC	FPGA	2 seconds
Slave	PTPv1	100 µs without PRC 10ns with PRC	FPGA	

4.2.2 Oscilloscope Specification

In the experimental study, an Agilent MSO7104A Mixed Signal Oscilloscope with firmware 06.16.0001 was used. This oscilloscope has the ability to measure the time difference between two analogue input signals. This oscilloscope has the capability of measuring the time difference from a selected edge between two sources [56] in our case: 1 PPS output from GM and 1 PPS output from slave. The oscilloscope can also calculate several statistics concerning this time difference, specifically Mean and Standard deviation.

Table 4-2 shows the configuration of the oscilloscope channels while measuring the data in this thesis.

Table 4-2: Channel configuration of the osciloscope

Vertical Division	Coupling	BW limit	Impedance	Probe	Trigger	Mode	Holdoff
1.00V	DC	Off	1M Ohm	10:1	Edge	Normal	60ns

4.2.3 Traffic generator

In this thesis project, the IxN2X application was used which is used by many service providers and network equipment manufactures for testing metro/edge platforms, core routers, and optical switches. It simulates large-scale subscriber behavior and is used for measurement of quality of service (QoS).

The IxN2X, release 6.13 SR2 traffic generator was used to generate the different traffic loads as per Table 3-5.

4.3 Experimental Scenarios

In this section the different scenarios that were used in the experiments in this thesis will be discussed in detail. In all of these scenarios, the switches were configured to do layer 3 forwarding via a VLAN with untagged ports. IP forwarding is enabled in all switches.

4.3.1 Scenario 0

In the first scenario, we study the PTP protocol while the GM and slave are directly connected, i.e., without any nodes between them. The configuration for this scenario is shown in Figure 4-2. The link bandwidth between GM and Slave are 100 Mbps in this scenario.

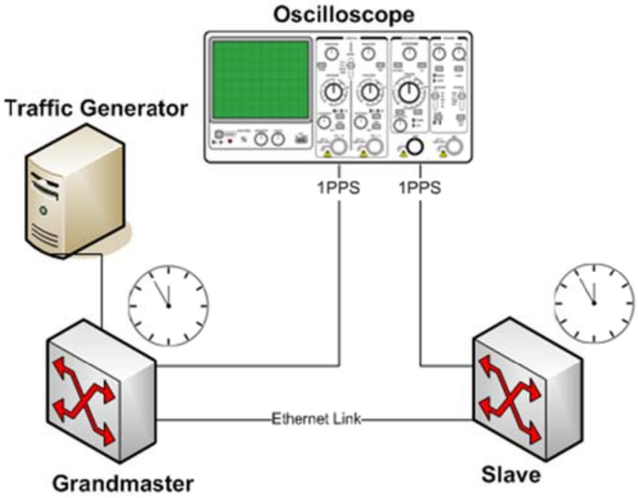

Figure 4-2: **Scenario 0 topology**

The traffic injected by the IXIA traffic generator was input to port two of the GM. This traffic consisted of link layer broadcast frames with different sizes in the different tests in this scenario. Broadcast frames were used as such a frame causes the switch to output the frame on every port of the switch, except for the source port. The detailed specification of the different traffic profiles were explained in Table 3-5.

In this scenario several tests were run to discover the effect of traffic load on the network when the GM and slave are directly connected. The six tests that were run are described in Table 4-3.

Table 4-3: **Scenario 0 test description**

Scenario 0 – Test 1	The delay between GM and slave measured while there is no traffic between them.
Scenario 0 – Test 2	The delay between GM and slave measured while Profile1 was injected to the GM. The maximum packet length used in this test was a 64 Byte layer 2 frame.
Scenario 0 – Test 3	The delay between GM and slave measured while Profile2 was injected to the GM. The maximum packet length used in this test was a 64 Byte layer 2 frame.
Scenario 0 – Test 4	The delay between GM and slave measured while Profile4 was injected to the GM. The maximum packet length used in this test was a 64 Byte layer 2 frame.
Scenario 0 – Test 5	The delay between GM and slave measured while Profile 5 was injected to the GM. The maximum packet length used in this test was a 64 Byte layer 2 frame.
Scenario 0 – Test 6	The delay between GM and slave measured while Profile 6 was injected to the GM. The maximum packet length used in this test was a 64 Byte layer 2 frame.

4.3.2 Scenario 1

In this scenario, the relative delay between GM and slave are measured when there is one node between them. The configuration for this scenario is shown in Figure 4-3. The link bandwidth between GM, Vendor A and Slave are 100Mbps in this scenario.

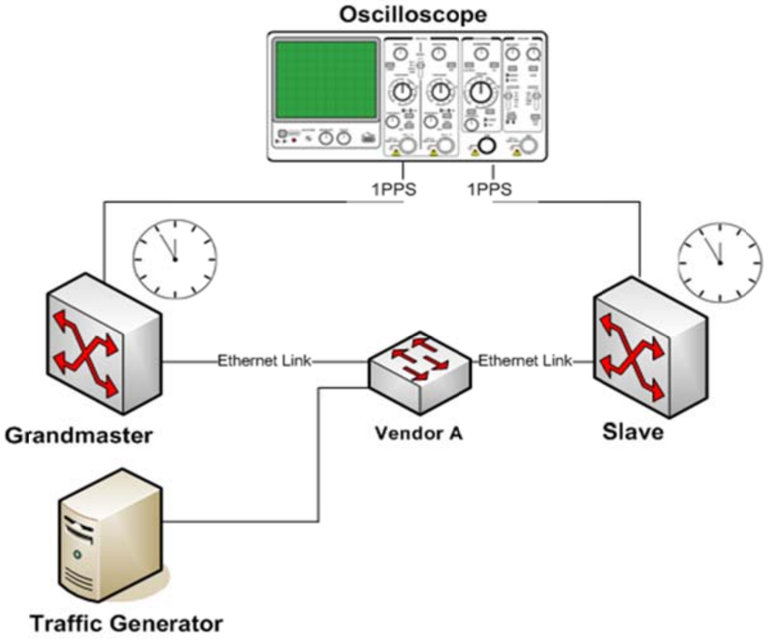

Figure 4-3: Scenario 1 topology

The traffic injected by IXIA traffic generator to the network is broadcast traffic with different sizes in different tests in this scenario. The traffic is a layer two frame. The detailed specification of the different traffic profiles were explained in Table 3-5.

In this scenario several tests run to discover the effect of traffic in the network between the GM and slave. The seven tests that were run are described in Table 4-4.

Table 4-4: Scenario 1 test description

Scenario 1 – Test 1	The delay between GM and slave measured while there is no traffic between them.
Scenario 1 – Test 2	The delay between GM and slave measured while traffic according to Profile1 was inject to the first node following the GM. The maximum packet length used in this test was a 64 Byte layer 2 frame.
Scenario 1 – Test 3	The delay between GM and slave measured while traffic according to Profile2 was injected to the first node following the GM. The maximum packet length used in this test was a 64 Byte layer 2 frame.
Scenario 1 – Test 4	The delay between GM and slave measured while traffic according to Profile3 was injected to the first node following the GM. The maximum packet length used in this test was a 64 Byte layer 2 frame.
Scenario 1 – Test 5	The delay between GM and slave measured while traffic according to Profile 4 was injected to the first node following the GM. The maximum packet length used in this test was a 64 Byte layer 2 frame.

Scenario 1 – Test 6	The delay between GM and slave measured while Profile 5 inject to the first node following the GM. The maximum packet length used in this test was a 64 Byte layer 2 frame.
Scenario 1 – Test 7	The delay between GM and slave measured while Profile 7 inject to the first node following the GM. The maximum packet length used in this test was a 1518 Byte layer 2 frame.
Scenario 1 – Test 8	The delay between GM and slave measured while Profile 8 inject to the first node following the GM. The maximum packet length used in this test was a 1518 Byte layer 2 frame.

4.3.3 Scenario 2

In this scenario, the relative delay between GM and slave was measured when there are two nodes between them. The configuration for this scenario is shown in Figure 4-4. The link bandwidth between GM and Slave in this scenario is shown in Table 4-5: Link bandwidth in scenario 2.

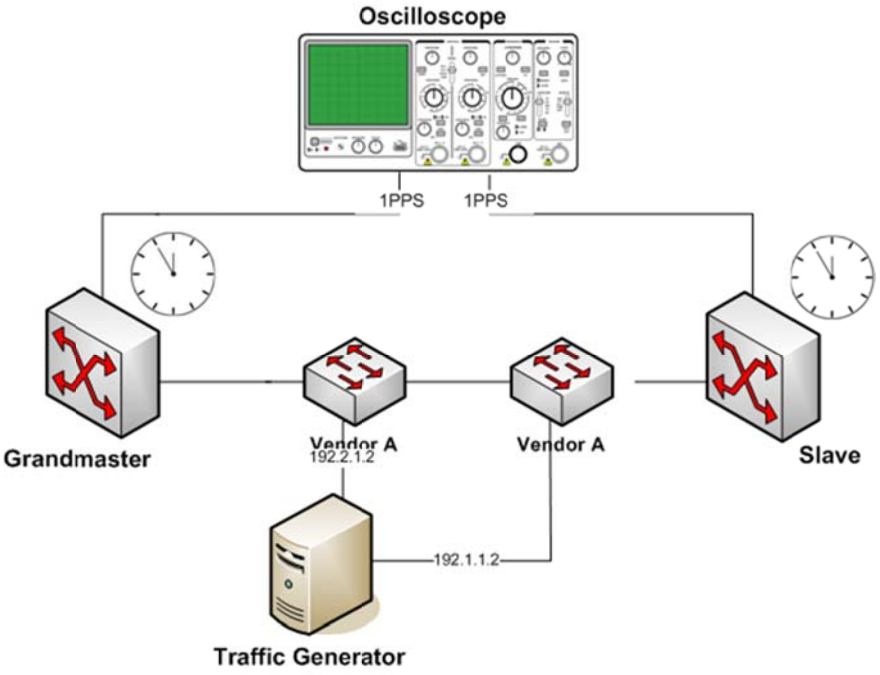

Figure 4-4: **Scenario 2 topology**

Table 4-5: **Link bandwidth in scenario 2**

	GM-Vendor A	**Vendor A – Vendor A**	**Vendor A – Slave**
Bandwidth	100 Mbps	1000 Mbps	100 Mbps

The traffic injected by one port of IXIA traffic generator with IP address 192.2.1.2 to another port of IXIA with IP address 192.1.1.2 with different sizes in different tests in this scenario. The traffic is a layer two frame. The detailed specification of the different traffic profiles were explained in Table 3-5.

In this scenario several tests run to discover the effect of traffic in the network between the GM and slave. The seven tests that were run are described in Table 4-6.

Table 4-6: Scenario 2 tests description

Scenario 2 – Test 1	The delay between GM and slave measured while there is no traffic between them.
Scenario 2 – Test 2	The delay between GM and slave measured while Profile2 inject to the GM. The maximum packet length used in this test was 64Byte with layer 2 frame.
Scenario 2 – Test 3	The delay between GM and slave measured while Profile3 inject to the GM. The maximum packet length used in this test was 64Byte with layer 2 frame.
Scenario 2 – Test 4	The delay between GM and slave measured while Profile4 inject to the GM. The maximum packet length used in this test was 64Byte with layer 2 frame.
Scenario 2 – Test 5	The delay between GM and slave measured while Profile 5 inject to the GM. The maximum packet length used in this test was 64Byte with layer 2 frame.
Scenario 2 – Test 6	The delay between GM and slave measured while Profile 6 inject to the GM. The maximum packet length used in this test was 64Byte with layer 2 frame.
Scenario 2 – Test 7	The delay between GM and slave measured while Profile 7 inject to the GM. The maximum packet length used in this test was 1518Byte with layer 2 frame.
Scenario 2 – Test 8	The delay between GM and slave measured while Profile 8 inject to the GM. The maximum packet length used in this test was 1518Byte with layer 2 frame.

4.3.4 **Scenario 3**

In this scenario, the relative delay between GM and slave are measured when there are three nodes between them. The configuration for this scenario is shown in Figure 4-5. The link bandwidth between GM and Slave in this scenario is shown in Table 4-7.

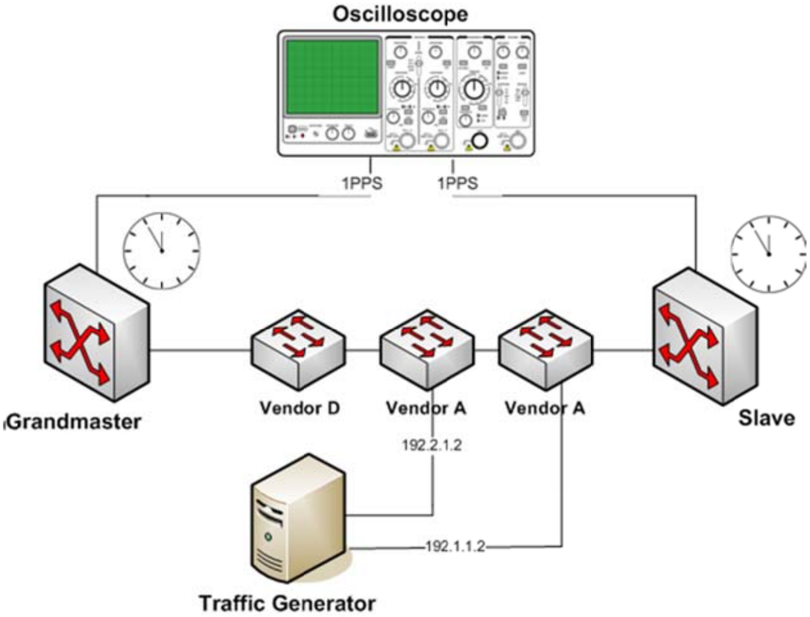

Figure 4-5: Scenario 3 topology

Table 4-7: Link Bandwidth in scenario 3

	GM-Vendor A	**Vendor D – Vendor A**	**Vendor A – Vendor A**	**Vendor A - Slave**
Bandwidth	100 Mbps	1000 Mbps	1000 Mbps	100 Mbps

The traffic injected by one port of IXIA traffic generator with IP address 192.2.1.2 to another port of IXIA with IP address 192.1.1.2 with different sizes in different tests in this scenario. The traffic is a layer two frame. The detailed specification of the different traffic profiles were explained in Table 3-3.

In this scenario several tests were run to discover the effect of traffic in the network between the GM and slave. The seven tests that were run are described in Table 4-8.

Table 4-8: Scenario 3 tests description

Scenario 3 – Test 1	The delay between GM and slave measured while there is no traffic between them.
Scenario 3 – Test 2	The delay between GM and slave measured while Profile1 inject to the GM. The maximum packet length used in this test was 64Byte with layer 2 frame.
Scenario 3 – Test 3	The delay between GM and slave measured while Profile2 inject to the GM. The maximum packet length used in this test was 64Byte with layer 2 frame.

Scenario 3 – Test 4	The delay between GM and slave measured while Profile3 inject to the GM. The maximum packet length used in this test was 64Byte with layer 2 frame.
Scenario 3 – Test 5	The delay between GM and slave measured while Profile 4 inject to the GM. The maximum packet length used in this test was 64Byte with layer 2 frame.
Scenario 3 – Test 6	The delay between GM and slave measured while Profile 5 inject to the GM. The maximum packet length used in this test was 64Byte with layer 2 frame.
Scenario 3 – Test 7	The delay between GM and slave measured while Profile 7 inject to the GM. The maximum packet length used in this test was 1518Byte with layer 2 frame.
Scenario 3 – Test 8	The delay between GM and slave measured while Profile 8 inject to the GM. The maximum packet length used in this test was 1518Byte with layer 2 frame.

4.3.5 **Scenario 4**

In this scenario, the relative delay between GM and slave are measured when there are four nodes between them. The configuration for this scenario is shown in Figure 4-6. The link bandwidth between GM and Slave in this scenario is shown in Table 4-9.

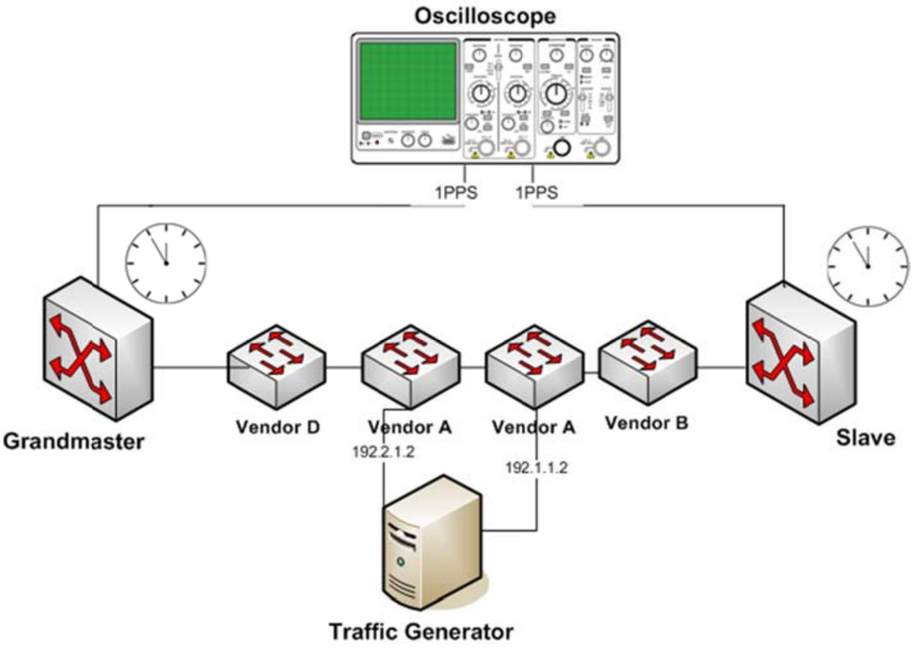

Figure 4-6: **Scenario 4 topology**

Table 4-9: Link Bandwidth in scenario 4

	GM- Vendor A	**Vendor D – Vendor A**	**Vendor A – Vendor A**	**Vendor A – Vendor B**	**Vendor B - Slave**
Bandwidth	100 Mbps	1000 Mbps	1000 Mbps	100 Mbps	100 Mbps

The traffic injected by one port of IXIA traffic generator with IP address 192.2.1.2 to another port of IXIA with IP address 192.1.1.2 with different sizes in different tests in this scenario. The traffic is a layer two frame. The detailed specification of the different traffic profiles were explained in Table 3 3.

In this scenario several tests run to discover the effect of traffic in the network between the GM and slave. The seven tests that were run are described in Table 4-10.

Table 4-10: Scenario 5 tests description

Scenario 4 – Test 1	The delay between GM and slave measured while there is no traffic between them.
Scenario 4 – Test 2	The delay between GM and slave measured while Profile1 inject to the GM. The maximum packet length used in this test was 64Byte with layer 2 frame.
Scenario 4 – Test 3	The delay between GM and slave measured while Profile2 inject to the GM. The maximum packet length used in this test was 64Byte with layer 2 frame.
Scenario 4 – Test 4	The delay between GM and slave measured while Profile3 inject to the GM. The maximum packet length used in this test was 64Byte with layer 2 frame.
Scenario 4 – Test 5	The delay between GM and slave measured while Profile4 inject to the GM. The maximum packet length used in this test was 64Byte with layer 2 frame.
Scenario 4 – Test 6	The delay between GM and slave measured while Profile 5 inject to the GM. The maximum packet length used in this test was 64Byte with layer 2 frame.
Scenario 4 – Test 7	The delay between GM and slave measured while Profile 7 inject to the GM. The maximum packet length used in this test was 1518Byte with layer 2 frame.
Scenario 4 – Test 8	The delay between GM and slave measured while Profile 8 inject to the GM. The maximum packet length used in this test was 1518Byte with layer 2 frame.

4.3.6 Scenario 5

In this scenario, the relative delay between GM and slave are measured when there are five nodes between them. The configuration for this scenario is shown in Figure 4-7. The link bandwidth between GM and Slave in this scenario is shown in Table 4-11: Link Bandwidth Scenario 5.

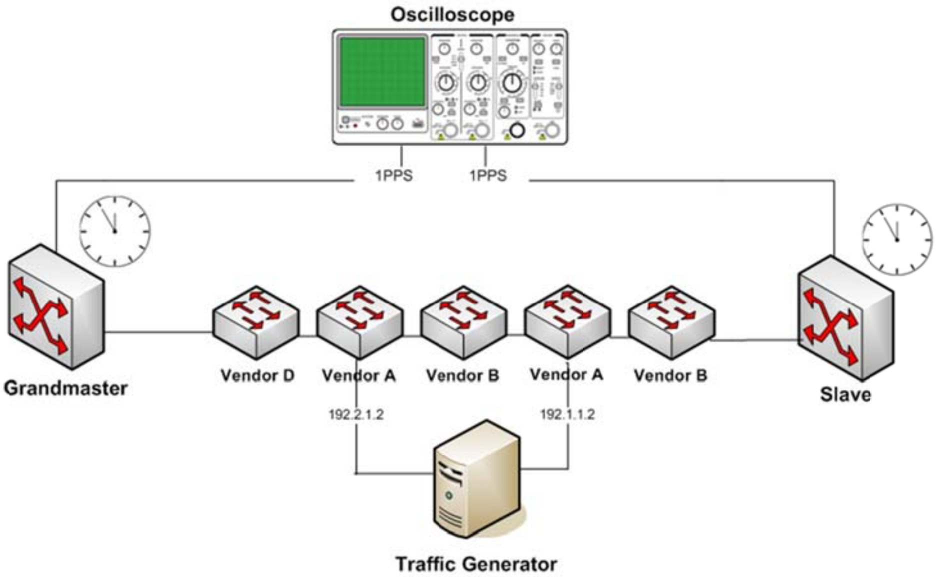

Figure 4-7: **Scenario 5 topology**

Table 4-11: **Link Bandwidth Scenario 5**

	GM-Vendor A	**Vendor D – Vendor A**	**Vendor A – Vendor A**	**Vendor A – Vendor B**	**Vendor B - Slave**
Bandwidth	100 Mbps	100 Mbps	100 Mbps	100 Mbps	100 Mbps

The traffic injected by one port of IXIA traffic generator with IP address 192.2.1.2 to another port of IXIA with IP address 192.1.1.2 with different sizes in different tests in this scenario. The traffic is a layer two frame. The detailed specification of the different traffic profiles were explained in Table 3 3.

In this scenario several tests run to discover the effect of traffic in the network between the GM and slave. The seven tests that were run are described in Table 4-12.

Table 4-12: **Scenario 5 tests description**

Scenario 5 – Test 1	The delay between GM and Slave measured while there is no traffic between them.
Scenario 5 – Test 2	The delay between GM and Slave measured while Profile1 inject to the GM. The maximum packet length used in this test was 64Byte with layer 2 frame.
Scenario 5 – Test 3	The delay between GM and Slave measured while Profile2 inject to the GM. The maximum packet length used in this test was 64Byte with layer 2 frame.

Scenario 5 – Test 4	The delay between GM and Slave measured while Profile3 inject to the GM. The maximum packet length used in this test was 64Byte with layer 2 frame.
Scenario 5 – Test 5	The delay between GM and Slave measured while Profile4 inject to the GM. The maximum packet length used in this test was 64Byte with layer 2 frame.
Scenario 5 – Test 6	The delay between GM and Slave measured while Profile 5 inject to the GM. The maximum packet length used in this test was 64Byte with layer 2 frame.
Scenario 5 – Test 7	The delay between GM and Slave measured while Profile 6 inject to the GM. The maximum packet length used in this test was 1518Byte layer 2 frame.
Scenario 5 – Test 8	The delay between GM and Slave measured while Profile 6 inject to the GM. The maximum packet length used in this test was 1518Byte layer 2 frame.

4.3.7 Scenario 6

In this scenario, the relative delay between GM and slave are measured when there are six nodes between them. The configuration for this scenario is shown in Figure 4-8. The link bandwidth between GM and Slave in this scenario is shown in Table 4-13.

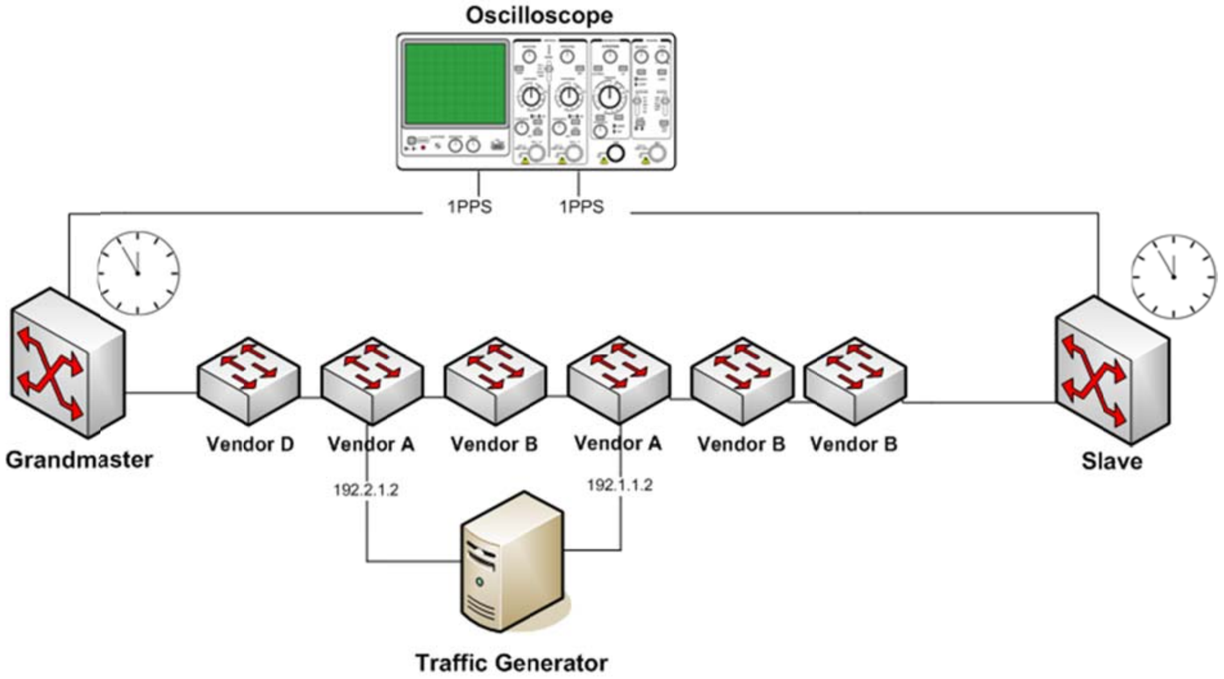

Figure 4-8: **Scenario 6 topology**

Table 4-13: **Link Bandwidth scenario 6**

	GM-Vendor A	**Vendor D – Vendor A**	**Vendor A – VendorB**	**Vendor B – Vendor A**	**Vendor B - Slave**
Bandwidth	100 Mbps	100 Mbps	100 Mbps	100 Mbps	100 Mbps

The traffic injected by one port of IXIA traffic generator with IP address 192.2.1.2 to another port of IXIA with IP address 192.1.1.2 with different sizes in different tests in this scenario. The traffic is a layer two frame. The detailed specification of the different traffic profiles were explained in Table 3 3.

In this scenario several tests run to discover the effect of traffic in the network between the GM and slave. The seven tests that were run are described in Table 4-14.

Table 4-14: Scenario 6 tests description

Scenario 6 – Test 1	The delay between GM and slave measured while there is no traffic between them.
Scenario 6 – Test 2	The delay between GM and slave measured while traffic according to Profile1 was injected to the third node following the GM. The maximum packet length that used in this test was a 64 Byte layer 2 frame.
Scenario 6 – Test 3	The delay between GM and slave measured while traffic according to Profile2 was injected to the third node following the GM. The maximum packet length that used in this test was a 64 Byte layer 2 frame.
Scenario 6 – Test 4	The delay between GM and slave measured while traffic according to Profile3 was injected to the third node following the GM. The maximum packet length that used in this test was a 64 Byte layer 2 frame.
Scenario 6 – Test 5	The delay between GM and slave measured while according to Profile4 was injected to the third node following the GM. The maximum packet length that used in this test was a 64 Byte layer 2 frame.
Scenario 6 – Test 6	The delay between GM and slave measured while traffic according to Profile 5 was injected to the third node following the GM. The maximum packet length that used in this test was a 64 Byte layer 2 frame.
Scenario 6 – Test 7	The delay between GM and slave measured while traffic according to Profile 7 was injected to the third node following the GM. The maximum packet length used in this test was a 1518 Byte layer 2 frame.
Scenario 6 – Test 8	The delay between GM and slave measured while traffic according to Profile 6 was injected to the third node following the GM. The maximum packet length used in this test was a 518 Byte layer 2 frame.

4.3.8 **Scenario 7**

In this scenario, the relative delay between GM and slave measured while there are seven nodes in the middle. The configuration for this scenario is shown in Figure 4-9. The link bandwidth between GM and Slave in this scenario is shown in Table 4-15.

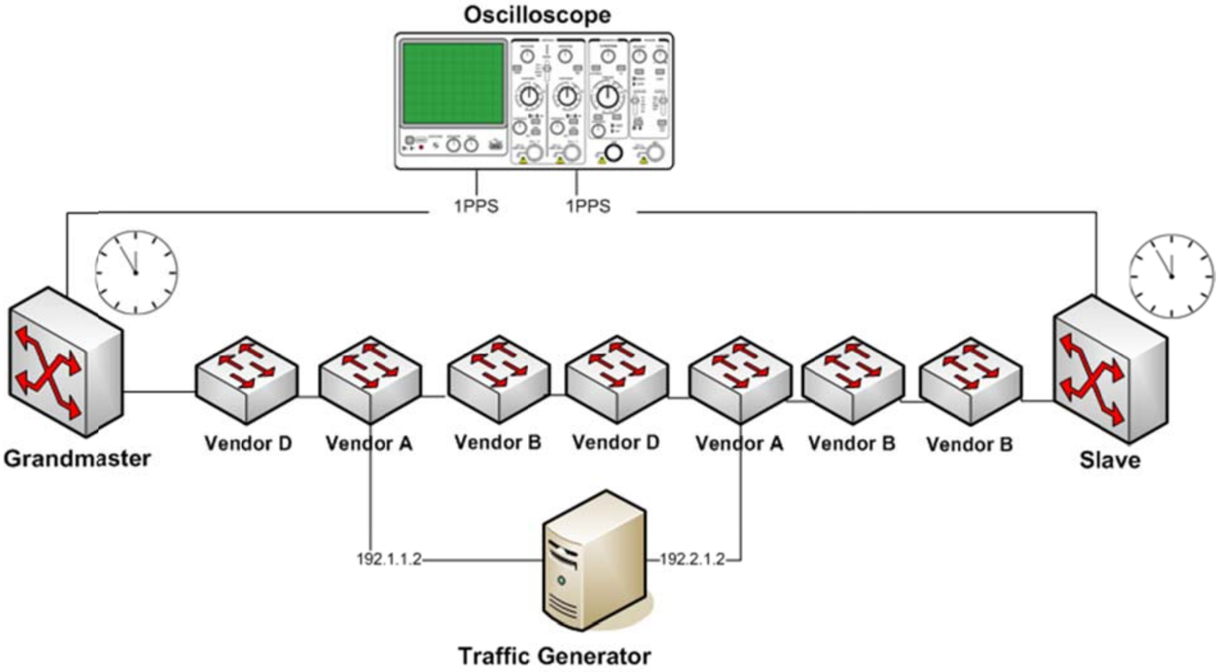

Figure 4-9: Scenario 7 topology

Table 4-15: Link Bandwidth scenario 7

	GM-Vendor A	**Vendor D – Vendor A**	**Vendor A – VendorB**	**Vendor B – Vendor D**	**Vendor D – Vendor A**	**Vendor A – Vendor B**	**Vendor B - Slave**
Bandwidth	100 Mbps	1000 Mbps	1000 Mbps	1000 Mbps	1000 Mbps	100 Mbps	100 Mbps

The traffic injected by one port of IXIA traffic generator with IP address 192.2.1.2 to another port of IXIA with IP address 192.1.1.2 with different sizes in different tests in this scenario. The traffic is a layer two frame. The detailed specification of the different traffic profiles were explained in Table 3 3.

In this scenario several tests run to cover the effect of running traffic in the network while GM and Slave have a direct connection. The eight tests that were run are described in Table 4-16.

Table 4-16: Scenario 7 tests description

Scenario 7 – Test 1	The delay between GM and slave measured while there is no traffic between them.
Scenario 7 – Test 2	The delay between GM and slave measured while traffic according to Profile1 was injected to the third node following the GM. The maximum packet length used in this test was a 64 Byte layer 2 frame.
Scenario 7 – Test 3	The delay between GM and slave measured while traffic according to

	Profile2 was injected to the third node following the GM. The maximum packet length used in this test was a 64 Byte layer 2 frame.
Scenario 7 – Test 4	The delay between GM and slave measured while traffic according to Profile3 was injected to the third node following the GM. The maximum packet length used in this test was a 64 Byte layer 2 frame.
Scenario 7 – Test 5	The delay between GM and slave measured while traffic according to Profile4 was injected to the third node following the GM. The maximum packet length used in this test was a 64 Byte layer 2 frame.
Scenario 7 – Test 6	The delay between GM and slave measured while traffic according to Profile 5 was injected to the third node following the GM. The maximum packet length used in this test was a 64 Byte layer 2 frame.
Scenario 7 – Test 7	The delay between GM and slave measured while traffic according to Profile 7 was injected to the third node following the GM. The maximum packet length used in this test was a 1518 Byte layer 2 frame.
Scenario 7 – Test 8	The delay between GM and slave measured while traffic according to Profile 8 was injected to the third node following the GM. The maximum packet length used in this test was a 1518 Byte layer 2 frame.

4.3.9 Scenario 8

In this scenario, the relative delay between GM and slave measured while there are ten nodes in the middle. The configuration for this scenario is shown in Figure 4-10: Scenario 10 topology. The link bandwidth between GM and Slave in this scenario is shown in Table 4-17.

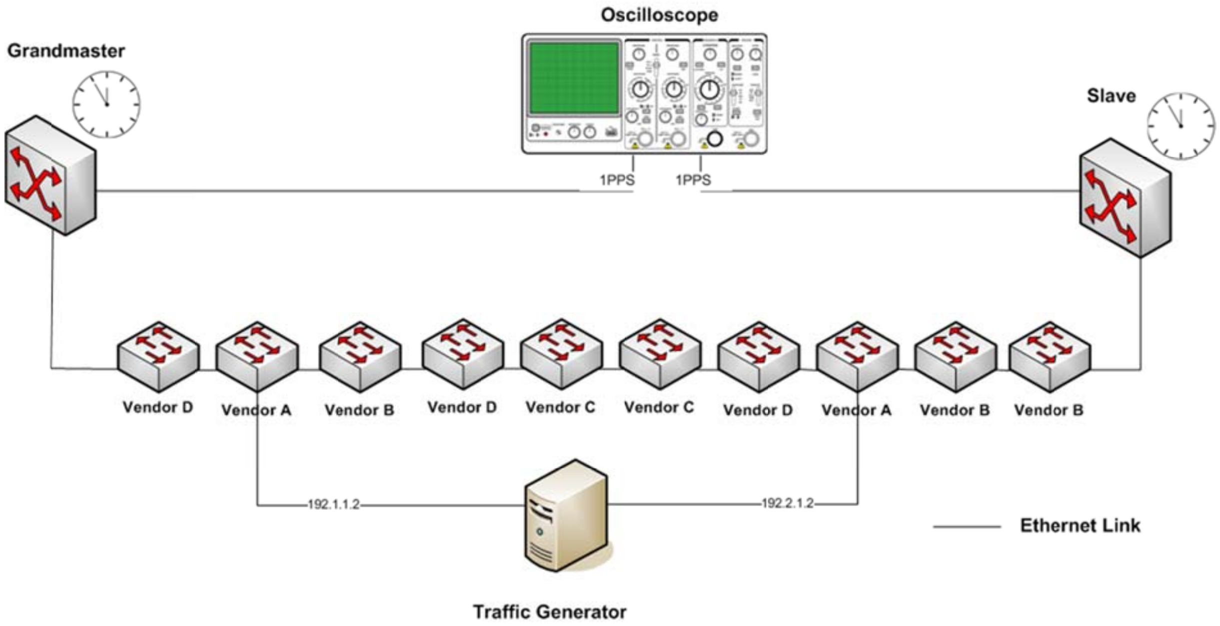

Figure 4-10: Scenario 10 topology

Table 4-17: Link Bandwidth scenario 8

	GM-A	**D –A**	**A –B**	**B –D**	**D –C**	**C –C**	**D –A**	**B –B**	**A –B**	**B - Slave**
Bandwidth	100 Mbps	1000 Mbps	1000 Mbps	1000 Mbps	100 Mbps	100 Mbps	1000 Mbps	100 Mbps	100 Mbps	100 Mbps

The traffic injected by one port of IXIA traffic generator with IP address 192.2.1.2 to another port of IXIA with IP address 192.1.1.2 with different sizes in different tests in this scenario. The traffic is a layer two frame. The detailed specification of the different traffic profiles were explained in Table 3-5.

In this scenario several tests run to cover the effect of running traffic in the network while GM and Slave have a direct connection. The eight tests that were run are described in Table 4-18.

Table 4-18: Scenario 8 tests description

Scenario 8 – Test 1	The delay between GM and slave is measured while there is no traffic between them.
Scenario 8 – Test 2	The delay between GM and slave is measured while traffic according to Profile1 was injected to the third node following the GM. The maximum packet length used in this test was a 64 Byte layer 2 frame.
Scenario 8 – Test 3	The delay between GM and slave is measured while traffic according to Profile2 was injected to the third node following the GM. The maximum packet length used in this test was a 64 Byte layer 2 frame.
Scenario 8 – Test 4	The delay between GM and slave is measured while traffic according to Profile3 was injected to the third node following the GM. The maximum packet length used in this test was a 64 Byte layer 2 frame.
Scenario 8 – Test 5	The delay between GM and slave is measured while traffic according to Profile4 was injected to the third node following the GM. The maximum packet length used in this test was a 64 Byte layer 2 frame.
Scenario 8 – Test 6	The delay between GM and slave is measured while traffic according to Profile 5 was injected to the third node following the GM. The maximum packet length used in this test was a 64 Byte layer 2 frame.
Scenario 8 – Test 7	The delay between GM and slave is measured while traffic according to Profile 7 was injected to the third node following the GM. The maximum packet length used in this test was a 1518 Byte layer 2 frame.
Scenario 8 – Test 8	The delay between GM and slave is measured while traffic according to Profile 8 was injected to the third node following the GM. The maximum packet length used in this test was a 1518 Byte layer 2 frame.

4.4 Emulating network characteristics using virtual bridge and Traffic Control

In this section the two scenarios that were used in the experiments to emulate network characteristics in this thesis will be discussed in detail. As it is obvious in Figure 4-11, GM and slave connected to each other using Linux Ubuntu which running virtual bridge. The detail specification of OS, CPU and memory used in this experiment is shown in Table 3-2.

In all of these scenarios, the PTP switches were connected to each other through virtual bridge (bridge utils package) in Ubuntu 12.4. In Figure 4-11 eth0 and eth1 are two interfaces of br0 (virtual bridge) and they forward any packet from one interface to another interface. This configuration is associated with scenario 0 that explained in 0 in which GM and slave are connected directly.

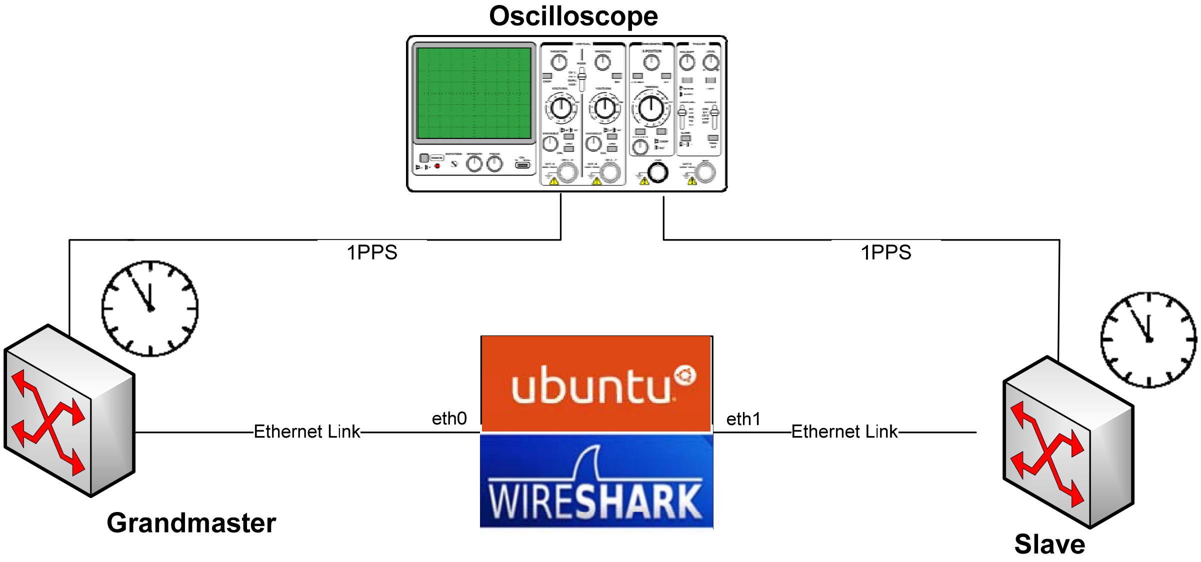

Figure 4-11: **GM and slave connected with virtual bridge**

It worth to note that the PTP packets are addressed to multicast address so, this virtual bridge should be able to forward multicast packets [57]. As multicast forwarding is no enabled in Linux Kernel by default, it is necessary to configure the Linux Kernel to support multicast routing. In this experiment PIM-SM daemon (pimd package) is used.

These are general points which are considered to enable multicast forwarding in Linux using virtual bridge:

- Ip forwarding is enabled.
- Build a kernel with MULTICAST is enabled (pimd is used in this thesis configuration).
- Virtual bridge interfaces is configured in promiscuous mode
- IGMP snooping is disabled [58].

4.4.1 Delay

In this scenario the effect of delay in the network is emulated. Different amount of delay is added to network by manually adding delay on the interface connected to slave using Traffic Control (TC) package in Linux. Table 4-19 shows the amount of delay that manually added to the network.

Table 4-19: Amount of delay manually added

Delay (microsecond)	Interface
10	Slave
100	Slave
200	Slave
500	Slave

4.4.2 Packet Loss

In this scenario the effect of PTP packet loss in the network is emulated. Traffic TC package in Linux is used to drop packets. Loss option in TC randomly drops packets on specified interface. To emulate packet burst losses, an optional correlation was used in this experiment which causes the random number generator to be less random. Table 4-20 explains the different profiles details used to emulate the packet loss.

Table 4-20: Profile ecplanation - Emulationg paclet loss

Profile	The delay between GM and slave measured while were connected to each other through virtual bridge. Different packet loss emulated using TC in Linux.
A	In this profile 10% loss added to interface connected to GM. No correlation was set so the packets dropped randomly.
B	In this profile 40% loss added to both interface connected to GM and Slave. No correlation was set so the packets dropped randomly.
C	In this profile 25% loss added to both interface connected to GM and Slave. 100% correlation was set to emulate bust traffic with 100 percentage load.
D	In this profile 25% loss added to both interface connected to GM and Slave. 100% correlation was set to emulate bust traffic with 100 percentage load.
E	In this profile 25% loss added to both interface connected to GM and Slave. 100% correlation was set to emulate bust traffic with 100 percentage load.

4.5 Data Collection

This section describes the instruments and methods used to collect the data. To achieve the goal of this thesis project we needed to measure the delay variation between the GM and slave in a packet switched network using scenarios designed to mimic the characteristics of a MAN. The GM device and slave device that were used in this thesis had a 1 PPS output. This output makes it possible to access the internal clock of these devices. The delay variation was collected by connecting the 1 PPS output of both the GM and slave to an oscilloscope.

In order to collect sufficient data to accurately measure both the delay variation and the drift (in this variation) over time each experimental test consisted of collecting data for at least 3600 seconds. This test duration was designed to give a confidence interval for our results of 95%. Given that the GM sent PTP messages at a rate of one every 2 seconds, at least 1800 PTP messages were emitted during each test run. The internal algorithm used by the slave to synchronize with the GM is unknown, hence we do not know how quickly the slave will discipline its local oscillator.

As mentioned previously we did not have a PRC connected to the GM, hence the GM's sense of time drifts with respect to UTC over the course of each experiment. Additionally, the oscilloscope was not connected to an external clock; hence its sense of time also drifts with respect to UTC.

5 Analysis

It this section, the metrics used to evaluate the results of this the experiments described in the previous chapter will be introduced. Moreover, the collected data were analyzed with R and the results of this analysis will be shown using different figures and then discussed.

5.1 Metrics

In order to evaluate the performance of PTP in a packet switched network under conditions comparable to a MAN (in this thesis this is based upon considering up to 10 switches on the path between the GM and slave, but the propagation delay along long physical links between these switches has not been considered), it is necessary to consider the delay characteristics of the network

5.1.1 Delay Variation

As discussed in section 2.1, different RBS technologies tolerate certain amounts of phase delay variation with regard to their required degree of synchronization. This thesis focused on phase synchronization, i.e., that the slave clock would remain within some specific range of difference from the GM in order to say that the slave was synchronized to the GM.

The main metric used in this project is time difference between the 1 PPS signal of the more accurate clock (assumed to be the GM) and the 1 PPS signal of the slave. As was explain in section 2.4 this metric is referred to as Time Error (TE).

5.1.2 Time Deviation (TDEV) / minTDEV

As was discussed in section 2.4.4, we will use Minimum Time Deviation (minTDEV) as a mask which is independent of the number of hops or switch time to evaluate whether the network meet the desired synchronization requirements or not.

5.2 Evaluation

In the first part of this section, the expected results from the experimental result will be discussed. In a subsequent part, the results of experimental study will be discussed.

5.2.1 Expected results

Increasing the number of nodes between the GM and the slave was expected to lead to an increase in the delay variation. The reason for this expectation is we assumed that the additional delay of the PTP messages passing through switch$_i$ will (in the absence of competing traffic) be independent of the additional delay introduced by switch$_j$, thus the delays are assumed to be independent. While each switch introduces increased delay (since a switch cannot introduce negative delay) the delay variation should be dependent upon the switch's architecture and realization. We assume that the delay and delay variation would be characteristics of a given vendor's make, model, and version of switch.

The results of increasing the traffic load on the slave's synchronization was expected to be little change in delay variance with a high constant load and directly related to the degree of burstiness with a highly bursty traffic load. Bursty traffic increases the scheduling processing of the switch and potentially also introduces a large output delay (as the probability that there are already packets to be output on a given port increases with increasing numbers of packets arriving before the PTP packets). Additionally, it takes longer for a switch to forward larger frames. Packet loss is another result of burstiness in the network.

When PTP switches experiencing fixed amount of delay, the delay variation will be stabled in about 30 to 130 seconds. A large amount of standard deviation when PTP packets are dropped in the network shows the instability of the network in terms of the means of delay (as shown in Table 5-2).

Section 5.2.1.1 explains the result of PDV in a emulated network. Section 5.2.1.2 explains the effects of packet loss by monitoring our experimented networks.

5.2.1.1 PTP Packet Delay Variation (PDV) in emulated network

Figure 5-1 shows the delay between GM and slave while they are connected via a virtual bridge. After adding 100 microsecond delay to interface connected to slave, the delay between GM and slave increased to reach 122 microsecond, then it gradually increased till got stabled around 47 microsecond. It took around 90 seconds that delay between GM and slave got stabled after adding 100 microsecond delay.

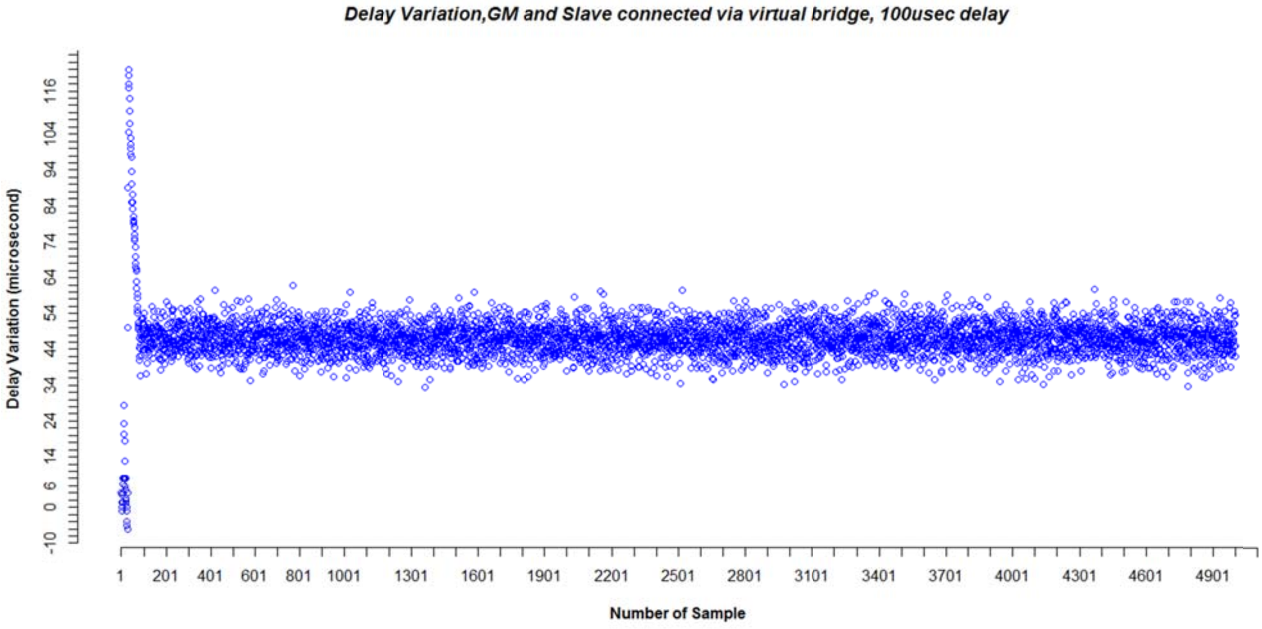

Figure 5-1: **Delay variation, 100usec delay added to network**

Figure 5-2 shows the delay variation while 10, 100, 200, and 500 microsecond delay is added to the interface which is connected to the slave. Figure 5-3 shows the absolute value of delay variation in those situations. According to the result of these measurement, the stabled amount of delay was about half of the delay added manually.

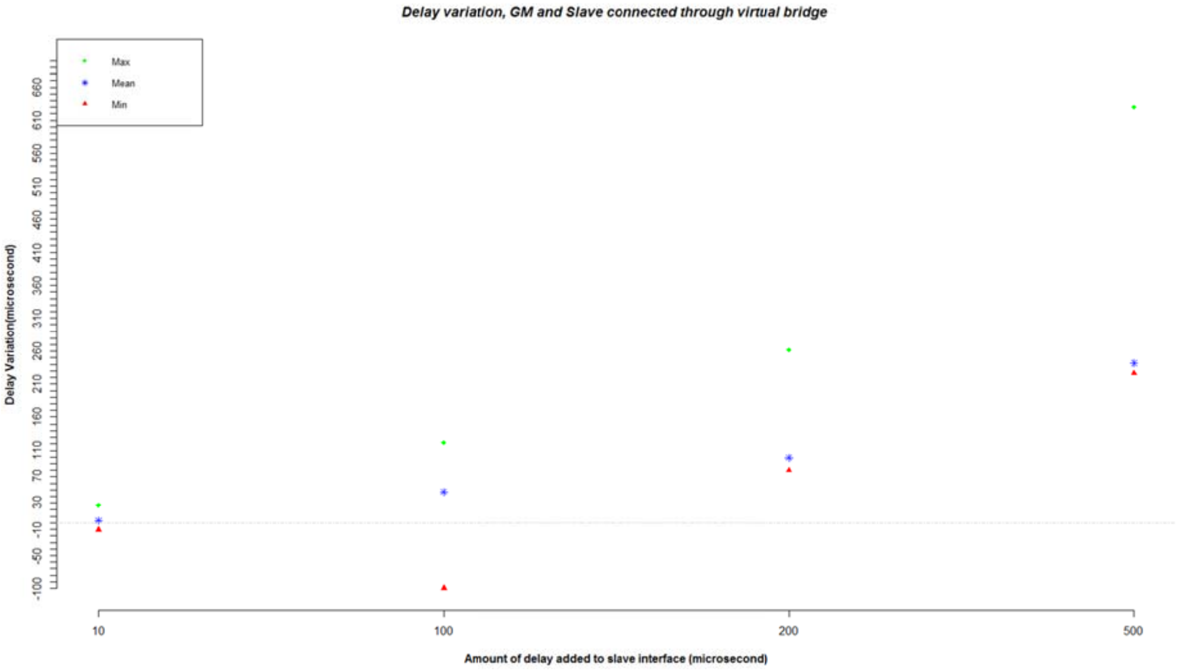

Figure 5-2: **Delay variation in different amount of delay added betwen GM and slave while connected through virtual bridge**

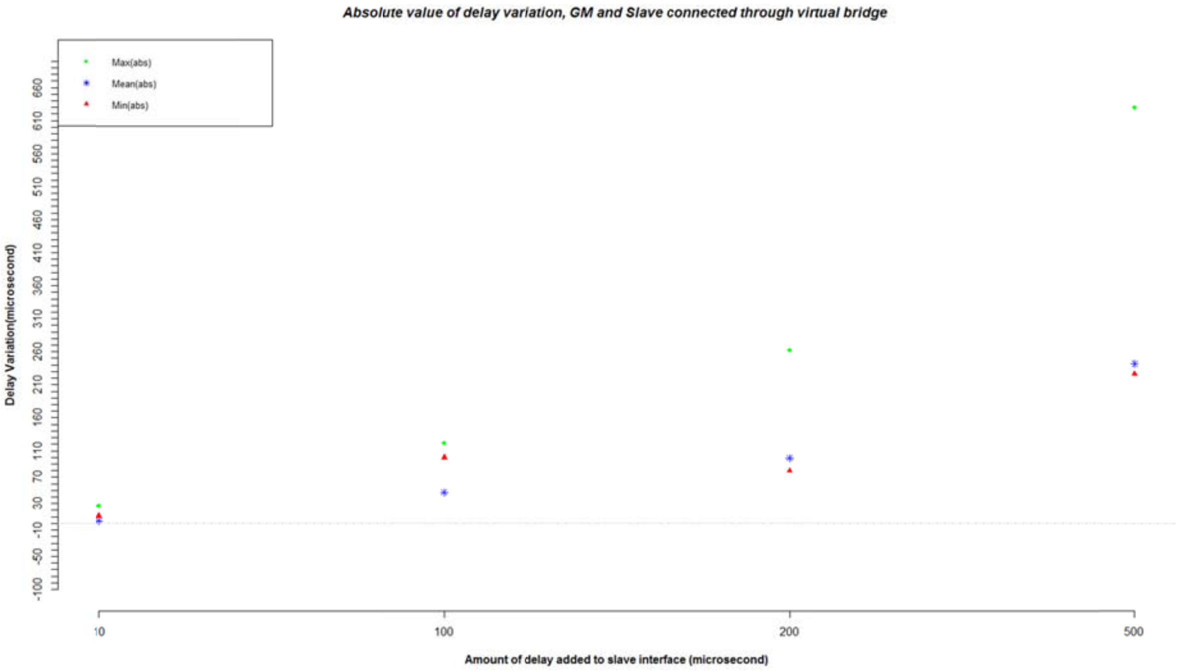

Figure 5-3: **Absolute value of delay variation between GM and slave while they connected together through virtual bridge**

Table 5-1 shows the standard deviation of delay variation between GM and slave when a different amount of delay was added to the network.

Table 5-1: Standard variation, Delay added while GM and slave connected through virtual bridge

	Delay manually added(microsecond)			
	10	100	200	500
Std Dev (µs)	2.3	4.17	12.1	13.9

Figure 5-4 shows the delay variation while different amounts of packet loss were added to both interfaces connected to GM and slave. Figure 5-5 shows the absolute value of delay variation between GM and slave while they are facing packet loss. Detail of each profile is explained in section 0.

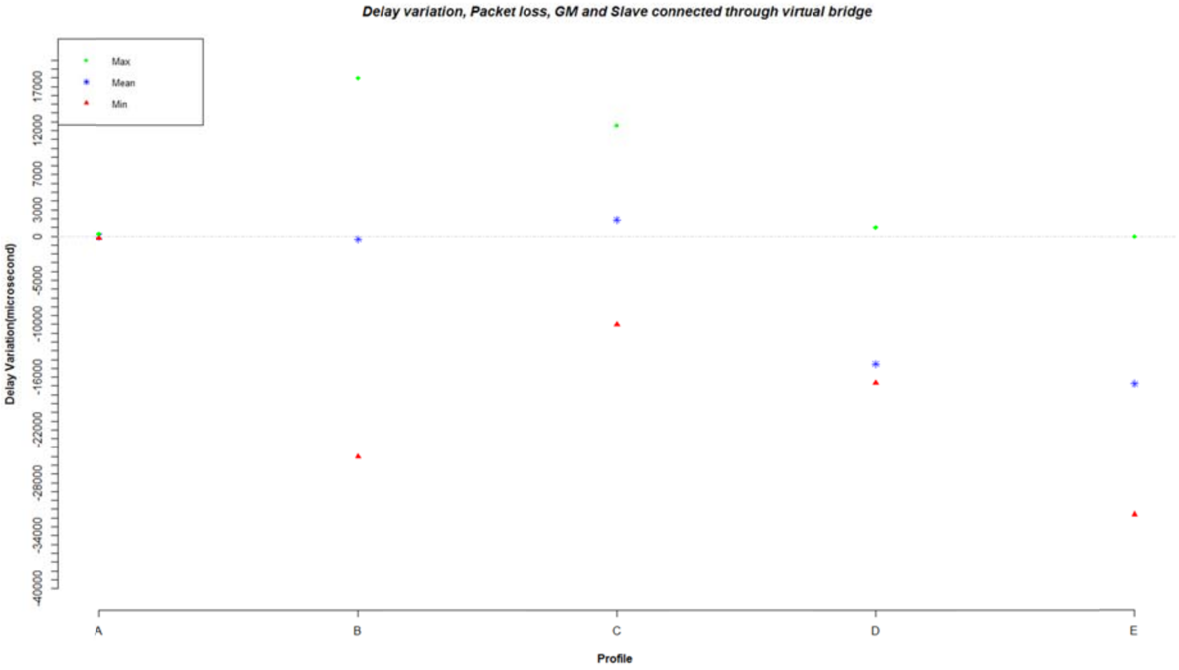

Figure 5-4: Delay variation, GM and Slave facing different amount of packet loss

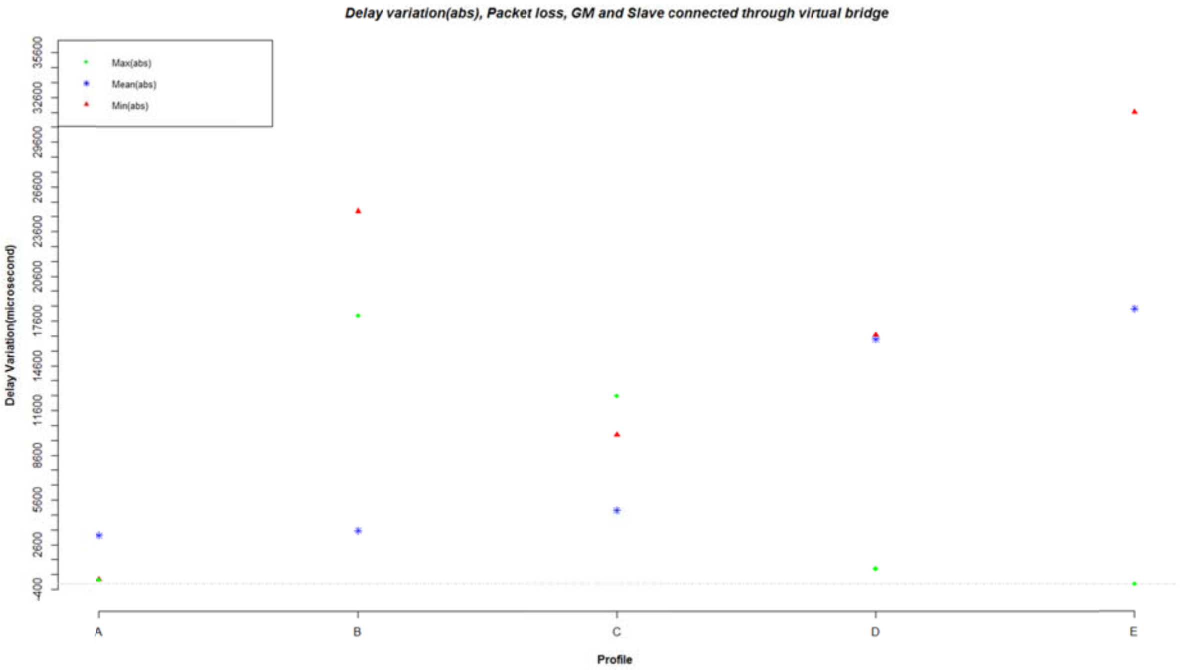

Figure 5-5: Abstract value of delay variation, GM and Slave facing different amount of packet loss

Table 5-2 shows the standard variation of delay variation between GM and slave while different percentages of packet loss. The large amount of standard variation shows the instability of the network in those situations.

Table 5-2: Standard variation in different profile, GM and slave connected through virtual bridge and facing packet loss

	Profile				
	A	B	C	D	E
Std Dev((µs))	50.7	1398	6470	5288	9740

5.2.1.2 Packet loss and PTP Packet Delay Variation (PDV)

Figure 5-6 shows the number of PTP lost packet in different profile when there is five nodes between GM and slave. PTP packets are monitored in different profiles from GM and slave. Result shows when there is no traffic running in the network around 1% of PTP packets get lost. While GM addressed to 75% of constant traffic load in the network around 4% of PTP packets get lost in the network. In profile 7, when switches subject to a very bursty load of 80% with maximum size of frame (1518 Byte), around 45% of PTP packets get lost in the network. In profile 8, switches subject to a full bandwidth load of traffic maximum size of frame (1518 Byte), around 95% of PTP packets get lost in the network.

It worth to mention that when switches does not subject to high amount of traffic, or even switches can handle large amount of traffic and slave receive the PTP packets we can expect that GM and slave got synchronized.

Figure 5-6: **Packet loss in different profiles, five nodes between GM and slave**

Figure 5-7 shows packet loss in profile 3 and 8 the network of scenario 5. It is obvious that when switches are subject to a full bandwidth load (100%) of traffic at maximum frame size (1518B), many of the packets get lost in the network because of switch congestion and long delay which confirms 95% of PTP packet loss in Figure 5-6 as a fraction of total packets in the network.

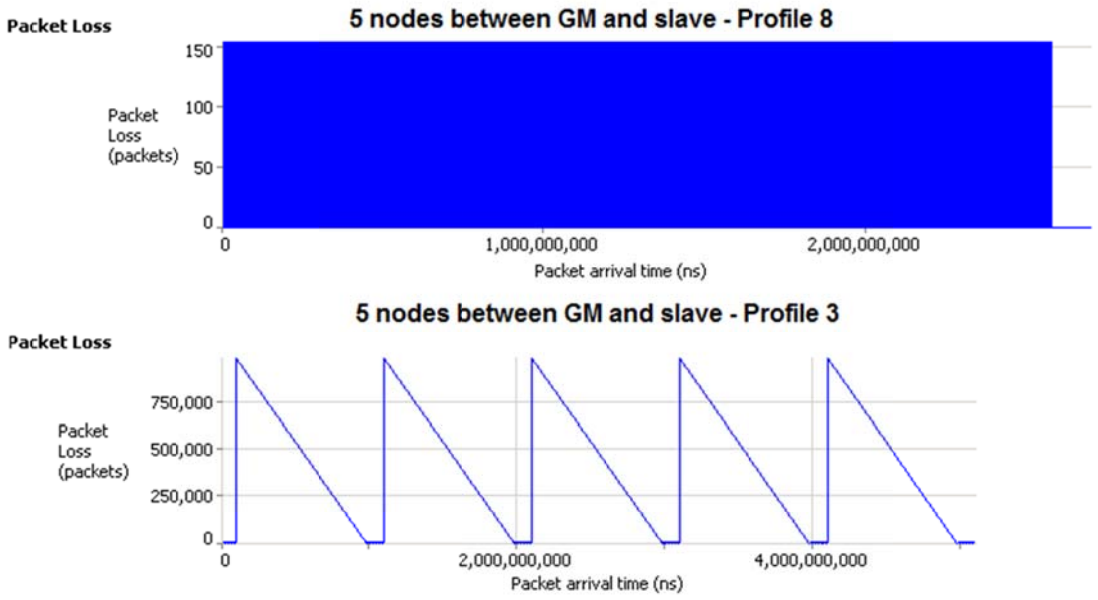

Figure 5-7: **Packet loss as captured by traffic generator in profiles 3 and 8, with 5 nodes between GM and slave**

Figure 5-8 shows PTP PDV when there are 5 nodes between GM and slave and no traffic running in the network. This figure is a result of 40,000 samples.

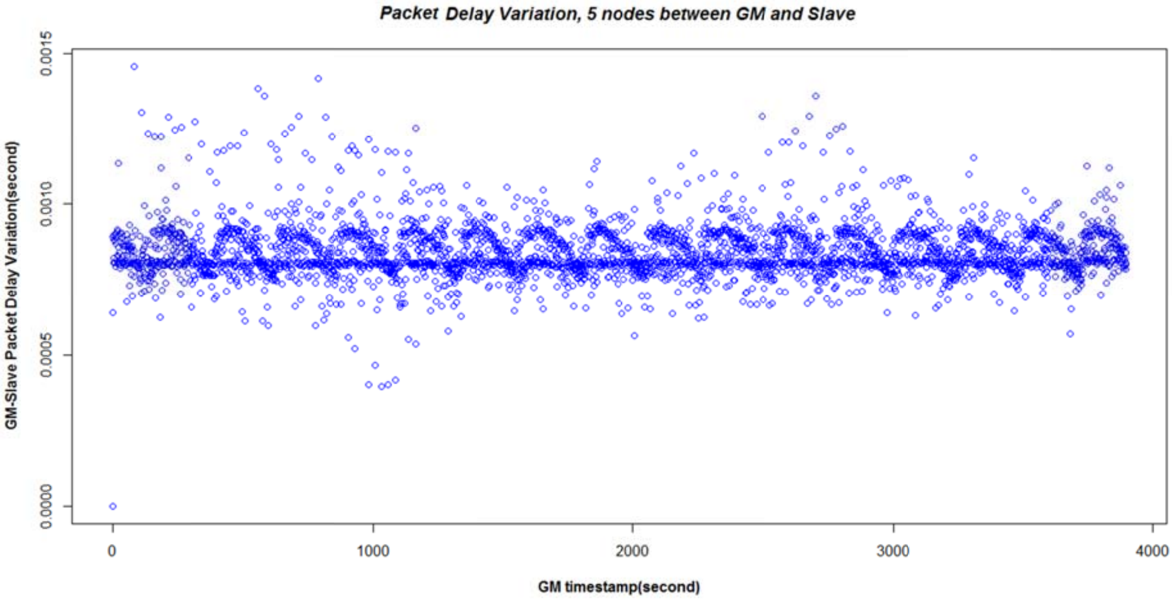

Figure 5-8: **PTP PDV, five nodes between GM and slave, no traffic in the network**

Figure 5-9 shows the histogram of PTP PDV when there are 5 nodes between GM and slave and no traffic running in the network. Ideally the distribution of the IP packet latency should be uniform distribution which fit to a Gaussian fit that has 13.44 microseconds wide. This is the time it takes to receiver 168 bytes (8 bytes of Ethernet header + 20 bytes of IP header + 8 bytes of UDP header + 132 bytes of UDP payload [59]. This fit shows in curve G2 in Figure 5-9. G1 is Gaussian curve based on all of the other delay associated with receiving the PTP packets and its variance. The final curve that fit the Gaussians based on the packet delay variation while five nodes between GM and slave is the sum of G1 and G2.

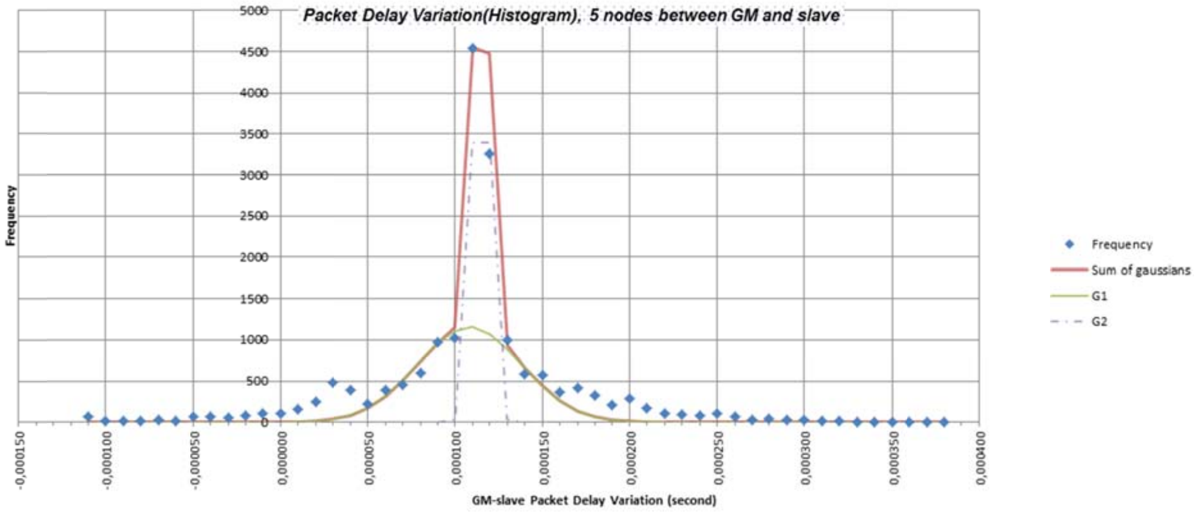

Figure 5-9: Histogram, five node between GM and slave, no traffic in the network

Figure 5-10 shows PTP GM and Slave versus relative time at sniffer while five nodes between GM and Slave and no traffic run in the network. Figure 5-11 shows delta PTP GM and Slave versus relative time at sniffer.

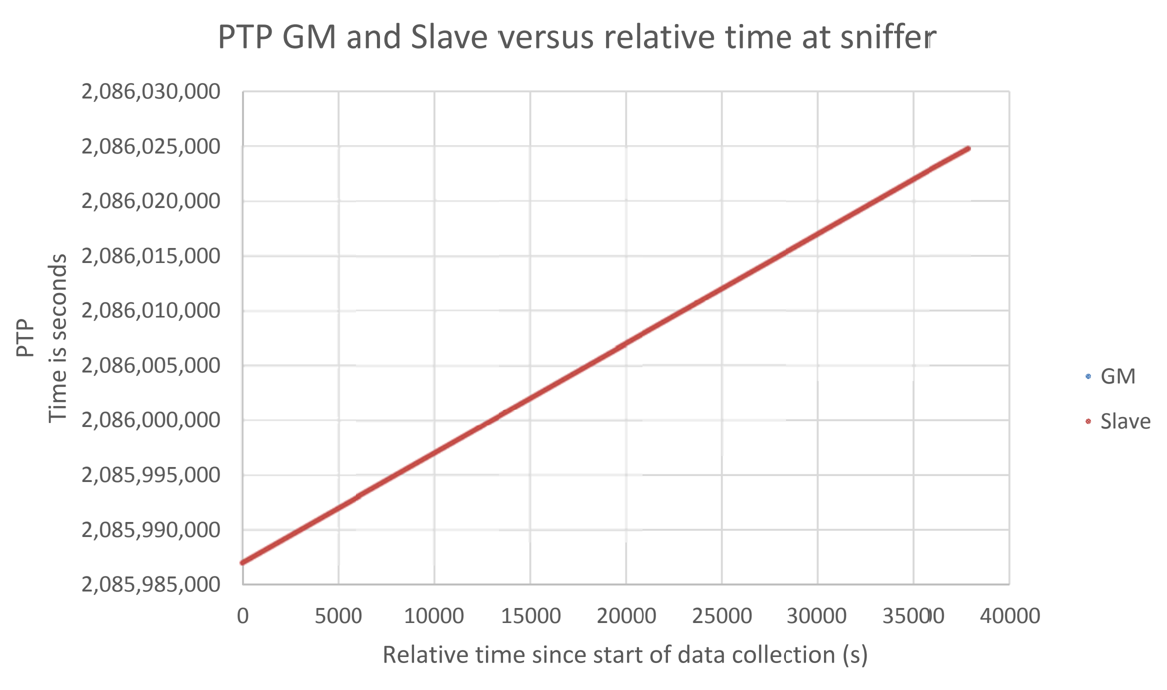

Figure 5-10: PTP GM and Slave versus relative time at sniffer, five node between GM and Slave, no traffic in the network

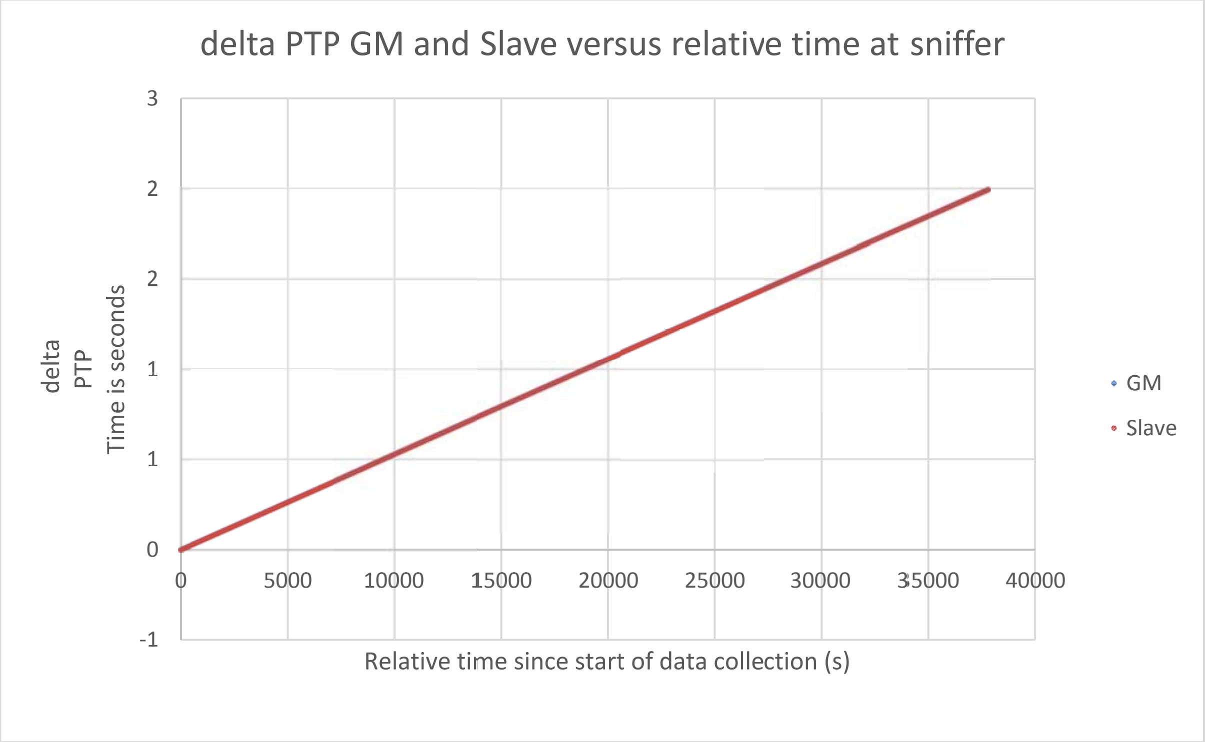

Figure 5-11: **Delta PTP GM and slave versus relative time at sniffer, five nodes between GM and Slave, no traffic in the network**

Figure 5-12 gives an overview of delta GM and Slave delay versus delta time.

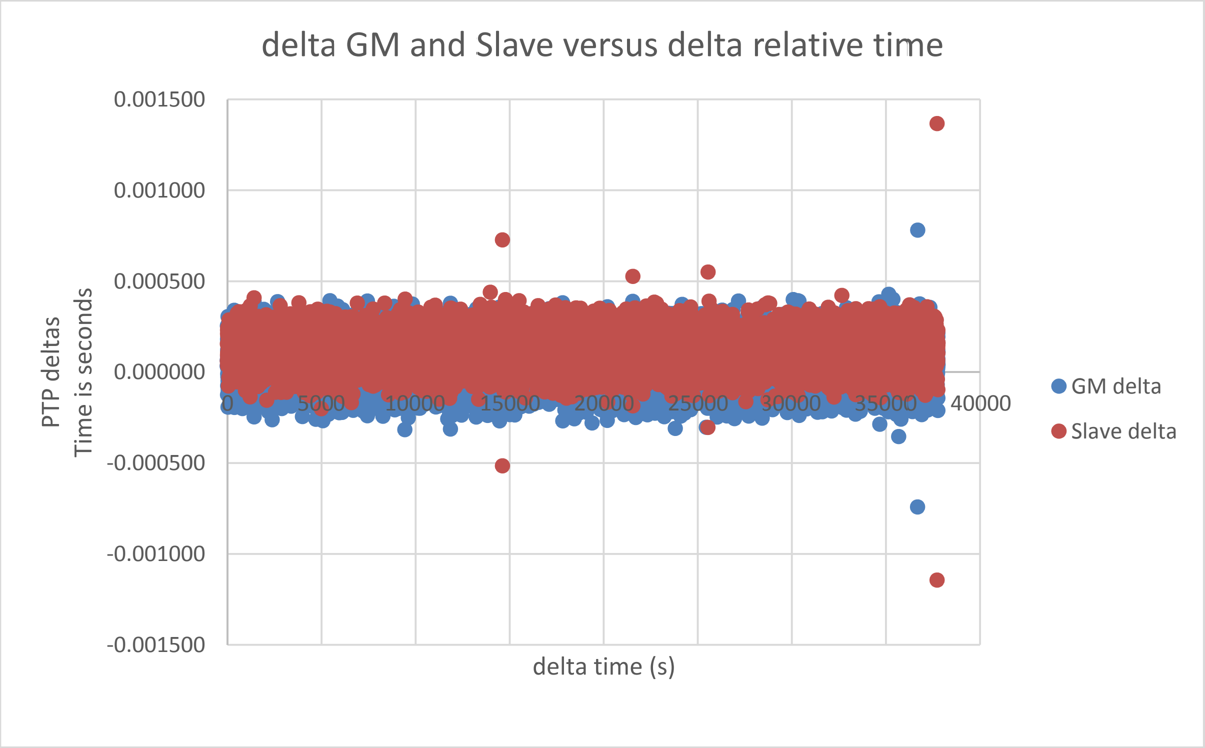

Figure 5-12: **Delta GM and slave versus delta relative time, five node between GM and Slave, no traffic in the network**

5.2.2 **Experimental Results**

The experimental results of an increasing number of nodes between the GM and slave support the expected result, as shown in the figures in the following paragraphs. The results are presented for the case of no traffic load and for several different traffic loads and patterns of load. The packet loss and PTP packet delay variation (PDV) for the different profiles when there are five nodes between master and slave will be discussed.

5.2.2.1 No Traffic

Figure 5-13 shows the minimum, mean, and maximum of delay variation between master and slave with different numbers of nodes (with a maximum of 10 nodes) in the path between the master and the slave over a period of 3 hours. The accuracy of oscilloscope in this measurement was set to 100 microseconds (µs), so the maximum delay it would capture is 499 microseconds. This curve shows the behavior when there is no other traffic in the network. When there is no node between master and slave, the delay is 0 µs, and master and slave are fully synchronized. The curve is approximately linear which confirms the theoretical expectation. As there is not much difference between minimum and maximum this indicates there is a stable clock.

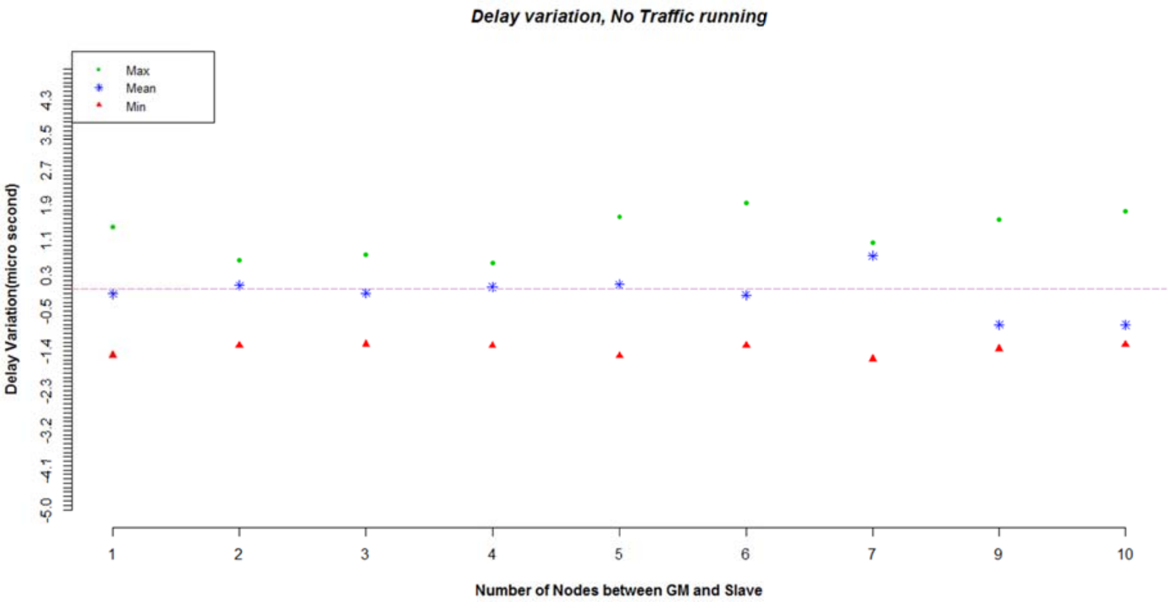

Figure 5-13: TE between GM and Slave while no traffic running

Figure 5-14 shows the absolute value of minimum, mean, and maximum of delay variation between master and slave with different numbers of nodes (with a maximum of 10 nodes) in the path between the master and the slave over a period of 3 hours. The maximum delay variation and minimum delay variation are approximately the same (the difference is tens of microseconds). Table 5-3 shows the standard deviation (Std. Dev.) for each situation. This low standard deviation shows there is low propagation delay variance in these situations, which confirms that the slave clock remains synchronized during these test runs.

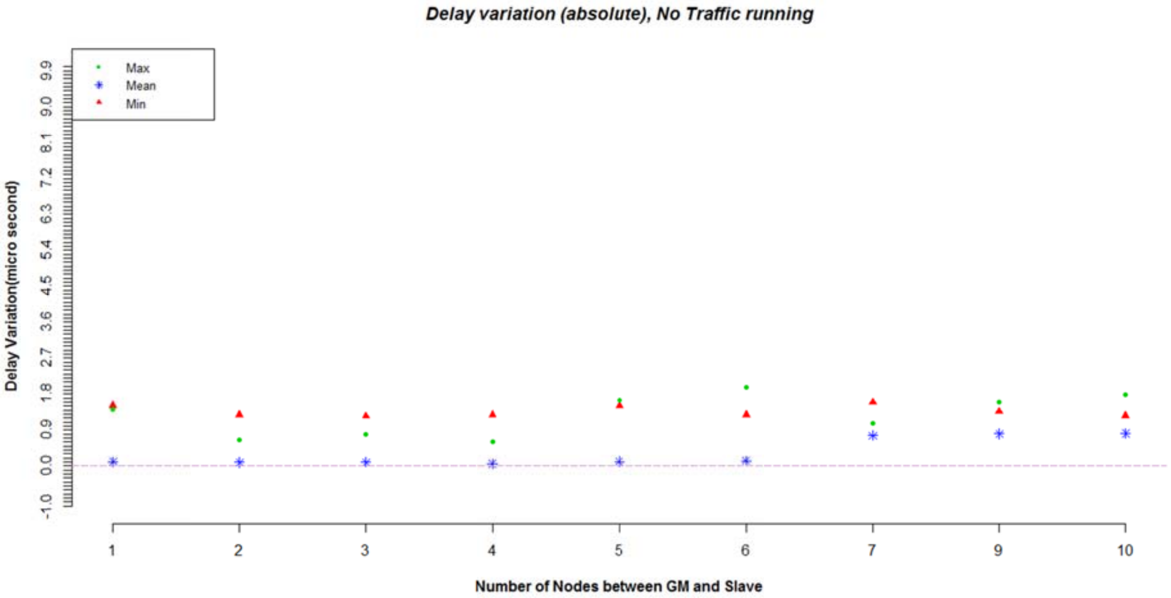

Figure 5-14: Absolute value of delay variation – No traffic running

Table 5-3: Standard Deviation

Number of nodes along the path	0	1	2	3	4	5	6	7	9	10
Std. Dev (µs)	0.075	1.410	0.650	0.780	0.595	1.650	1.960	1.065	1.590	1.780

As it mentioned in section 3.4.2, four different vendors of switches were used between the GM and slave for the experimental study in this thesis. Moreover, it is mentioned in section 3.1.1 that each switch's design features, such as internal scheduling algorithm and hardware design, affects the delay and delay variance it introduces in the network. See Table 5-4 for the minimum, maximum, and mean increase in delay variance for these four switches based upon 1200 samples.

Table 5-4: Delay Variance in different vendors

Increase in delay variance (in µs)

Vendor ID/Switch ID	Minimum	Mean	Maximum	Standard deviation (in µs)
A	-1	-0.295	0	0.255
B	-0.5	-0.185	0	0.946
C	-1.5	-0.349	19	0.616
D	-0.5	0.105	1	0.303

Table 5-5 shows test topologies based on the switch vendors that were shown in Figure 5-13. For all of the experiments the same topology was used for each of the different tests.

Table 5-5: Test topologies baased on number of nodes

1 Node	2 Nodes	3 Nodes	4 Nodes	5 Nodes	6 Nodes	7 Nodes	9 Nodes	10 Nodes
A	D-A	D-A-A	D-A-A-B	D-A-B-A-B	D-A-B-A-B-B	D-A-B-D-A-B-B	D-A-B-D-C-C-A-B-B	D-A-B-D-C-C-D-A-B-B

As was mentioned in section 2.3.2, the delay computation of the PTP protocol is based upon the master-to-slave time and slave-to-master time. The slave-to-master time is calculated using the delay_req message and delay_resp messages. Figure 5-15 shows the minimum, mean, and maximum of delay variation between master and slave with different numbers of nodes when the delay_req message was generated by one the middle nodes. In this case PTP works well, with the GM and slave synchronized even with 10 nodes between them.

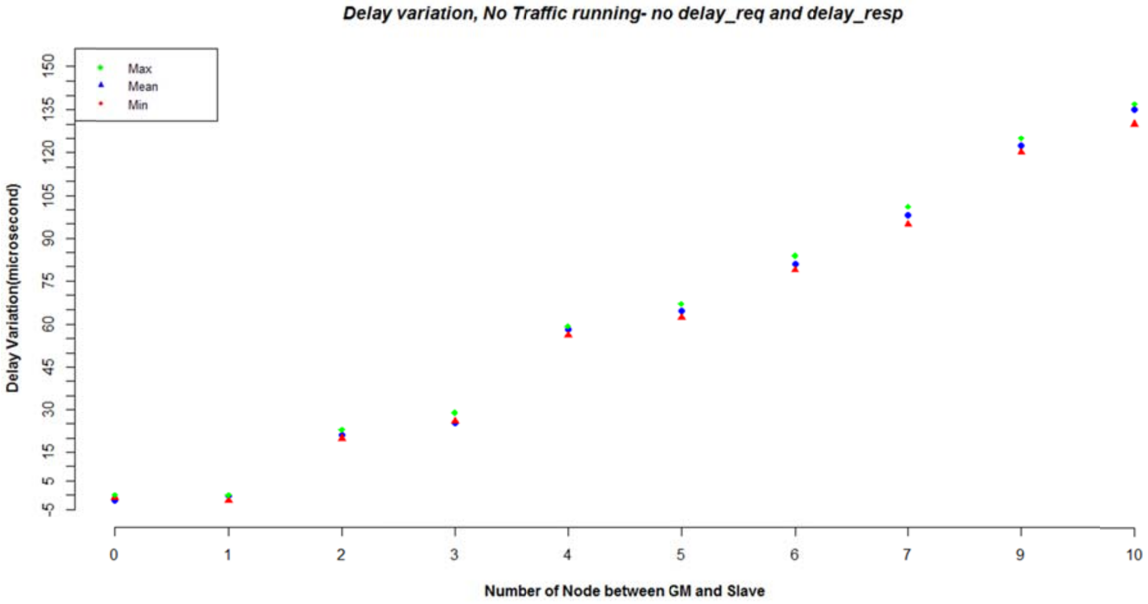

Figure 5-15: Delay varioation when there is no traffic in the networks, no delay_req and delay_resp

5.2.2.2 Scenario 0

Figure 5-16 shows the delay variation with different traffic loads when the GM and slave are directly connected. The mean delay variance in all load profiles was about 0 µs which means that the GM and slave are fully synchronized.

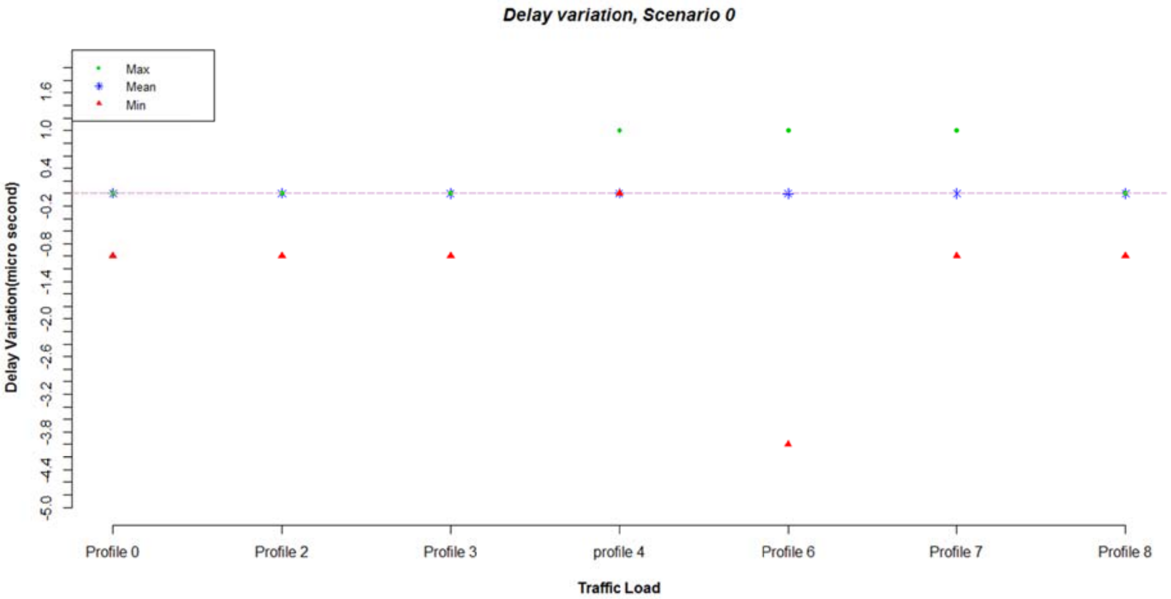

Figure 5-16: Delay Variation in Different Profile load, No node between GM and Slave

Figure 5-17 shows the absolute value of delay variation in scenario 0. The rather high standard deviation in profile 6 confirms a large difference between maximum and minimum delay variation in profile 6 in comparison to other profiles. Table 5-6 shows the standard deviation of the delay variation. The high standard deviation of the delay variance in the case of traffic profile 4 confirms the anomaly in the delay variance shown in Figure 5-16. Note that in this case the GM is subject to a very bursty load of 80% with 64 byte broadcast frames. This kind of traffic will take a lot of input and scheduling processing when these bursts of traffic appear, as contrasted with profile 6 which is just as bursty – but the packets are larger, hence the start of the packets are separated by more time.

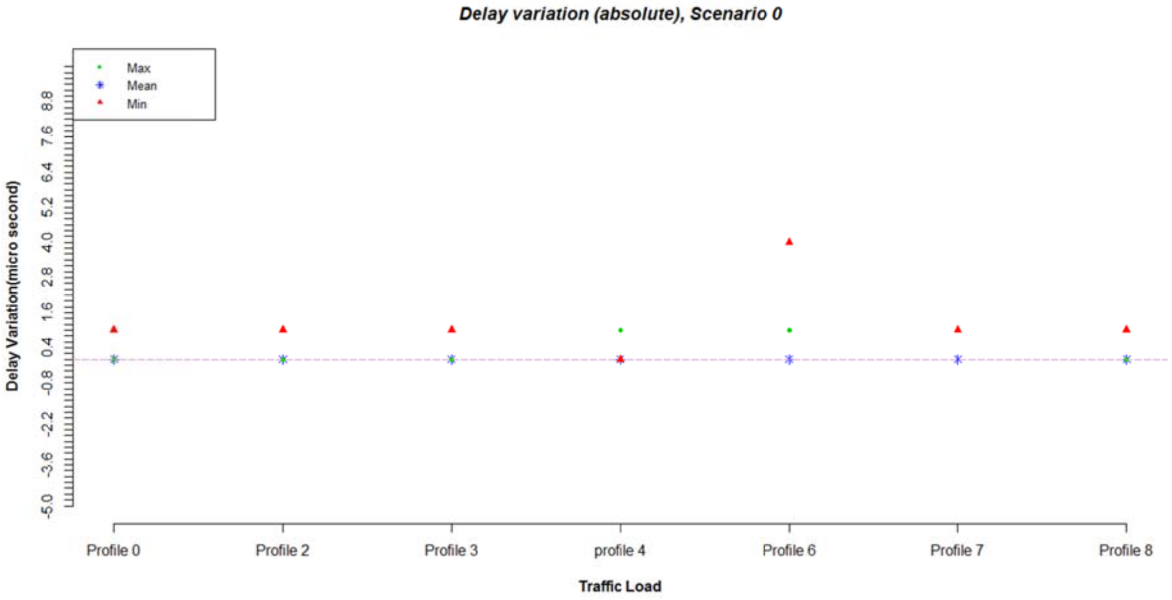

Figure 5-17: Absolute value of delay variation in scenario 0

Table 5-6: Standard deviation of delay variation with different load profiles- Scenario 0

	Profile						
	0	2	3	4	6	7	8
Std Dev (μs)	0.075	0.023	0.038	0.001	2.160	0.046	0.038

5.2.2.3 Scenario 1

Figure 5-18 shows the delay variation with different traffic load when there is one node between GM and. The mean delay variance in the different load profiles are between -14 and 4 μs which is acceptable synchronization.

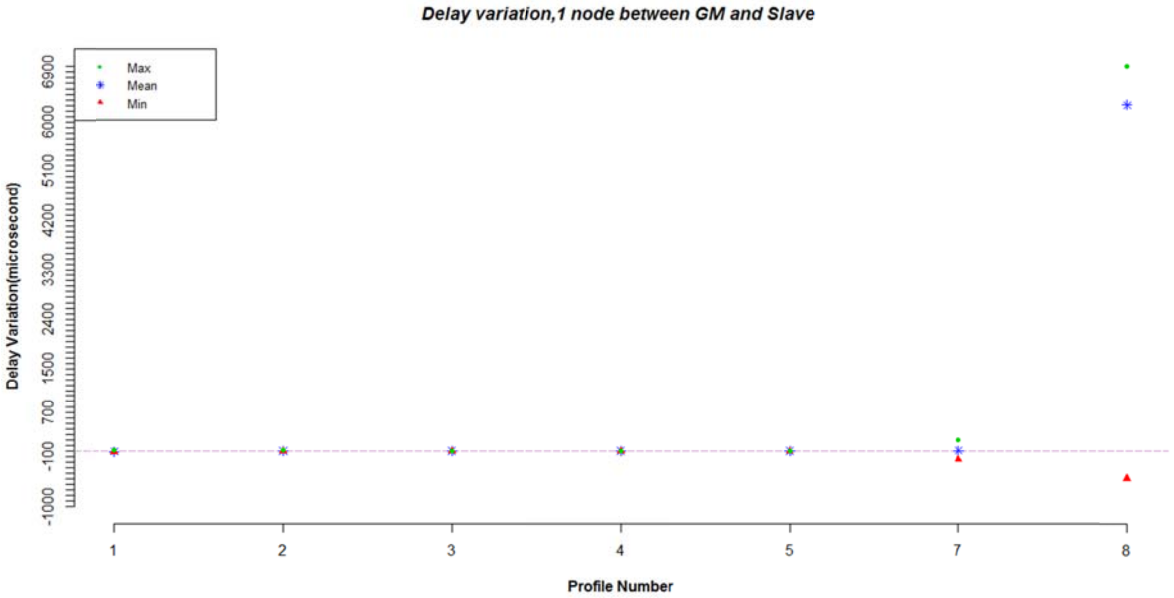

Figure 5-18: **Delay Variation in Different Profile load, one node between GM and Slave**

Figure 5-19 shows the absolute value of delay variation while there is one node between GM and slave. Table 5-7 shows the standard variation of delay variation. The high standard deviation of the delay variance in the case of traffic profile 8 confirms the anomaly in the delay variance shown in Figure 5-18. Note that in this case the GM is subject to a full bandwidth load (100%) of traffic with maximum size frames (1518 byte). It is visible in the Figure 5-4 that although profile 4 and profile 8 both carry 100% load in the network, Profile 4 contains 64 byte frames in contrast to Profile 8 that contains 1518 byte frame. Due to the larger frame size the time to transmit each frame is longer.

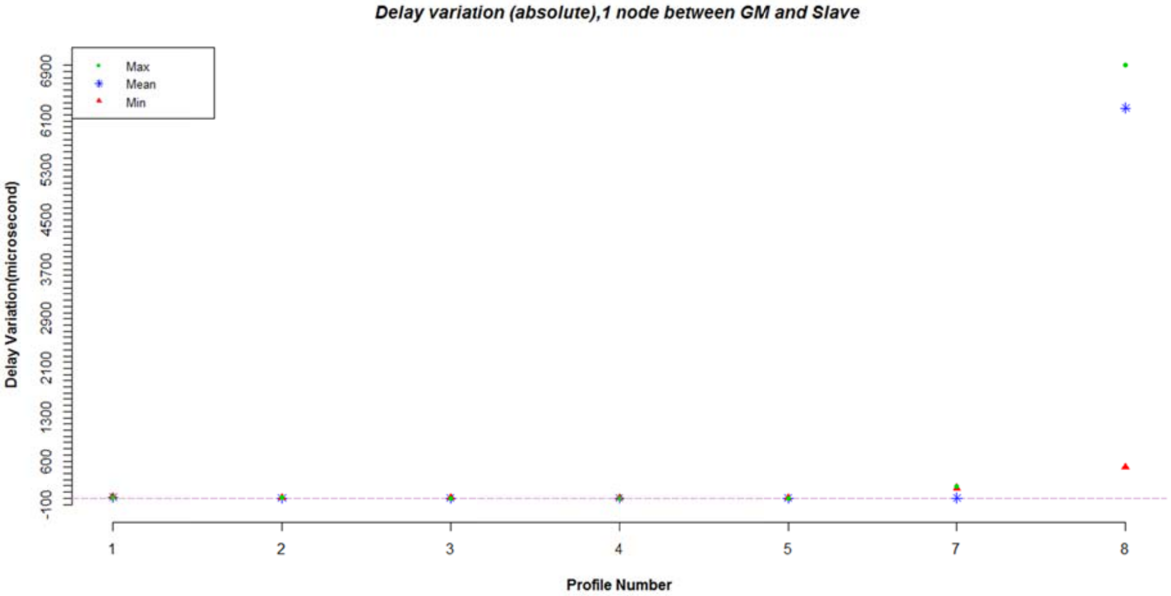

Figure 5-19: **Absolute value of delay variation in different profile load, one node between GM and Slave**

Table 5-7: **Standard deviation of delay variation with different load profiles, Scenario 1**

	Profile						
	1	2	3	4	5	7	8
Std Dev (µs)	22.209	3.344	1.045	0.588	2.255	42.45	708.83

5.2.2.4 Scenario 2

Figure 5-20 shows the delay variation with different traffic load when there is two nodes between GM and. The mean delay variance in different load profiles while the network having continues traffic with 64Byte frame are between -7 and 0 µs which is acceptable synchronization. However, the delay variation increases when traffic is bursty or has a large frame size.

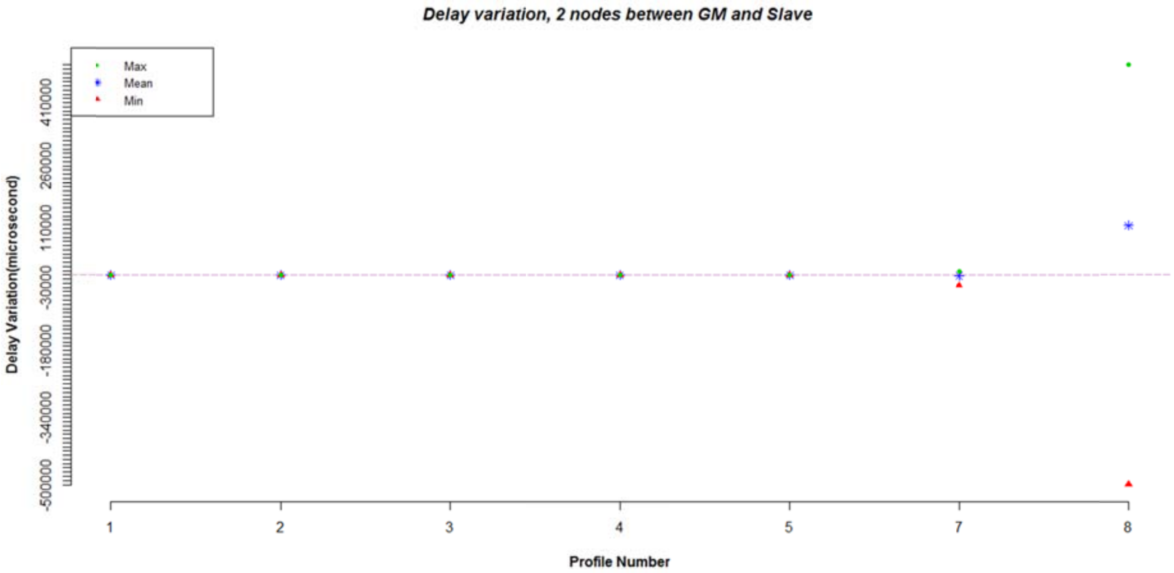

Figure 5-20: **Delay Variation in Different Profile load, two node between GM and Slave**

Figure 5-21 shows the absolute value of GM and slave.

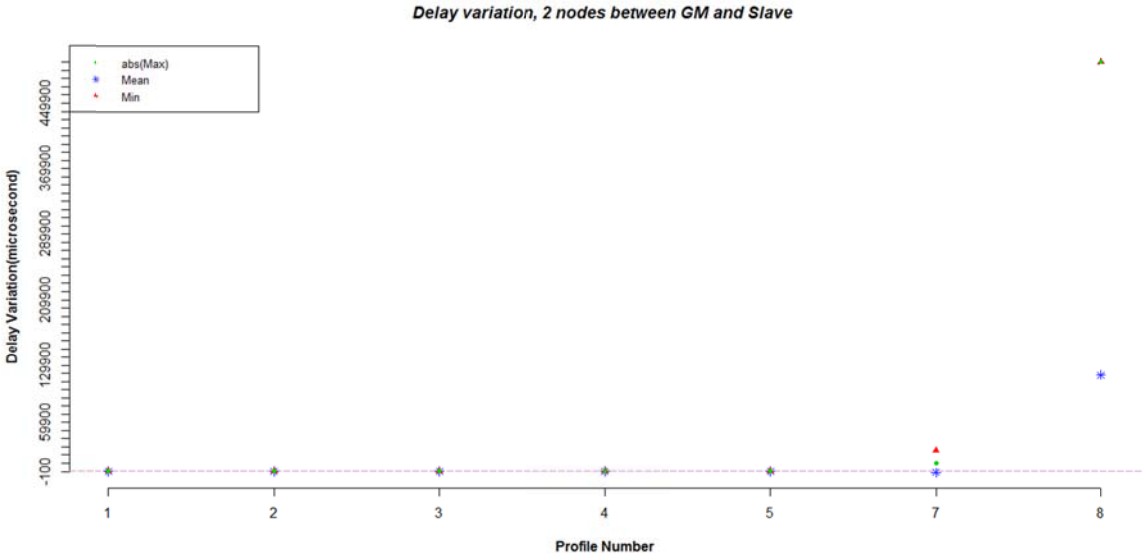

Figure 5-21: Absolute value of delay variation, two nodes between GM and Slave

Table 5-8 shows the standard deviation of delay variation. The high standard deviation of the delay variance in the case of traffic profile 8 confirms the anomaly in the delay variance shown in Figure 5-20. Note that in this case the GM is subject to a full bandwidth load (100%) of traffic with maximum size frames (1518 byte). As it mentioned in section 5.2.1.2, in this case many of PTP packets are lost which causes a huge delay in passing information between the GM and slave.

It is also visible that in profiles 5 and 7 which utilize bursty traffic, the standard deviation is rather high. The mean delay also is unacceptable for synchronization. Bursty traffic will take a lot of input and scheduling processing when these bursts of traffic appear which causes a larger variation in delays in contrast to profile loads with a constant traffic load. Moreover, packet loss is happening while switches address to bursty traffic which cause instability of delay variation.

Table 5-8: Standard deviation of delay variation with different load profiles, Scenario 2

	Profile						
	1	2	3	4	5	7	8
Std Dev (µs)	14.75	6897	0.852	1.388	2399	4700	218000

5.2.2.5 Scenario 3

Figure 5-22 shows the delay variation with different traffic load when there are three nodes between GM and slave. The mean delay variance in different load profiles while the network having continues traffic with 64Byte frame are between are between -13 and 9 µs which is acceptable synchronization. Delay variation in profile 5 which contains bursty traffic with 64Byte frame is -67 µs. Although is rather high in comparison to profiles 1- to 4 but acceptable.

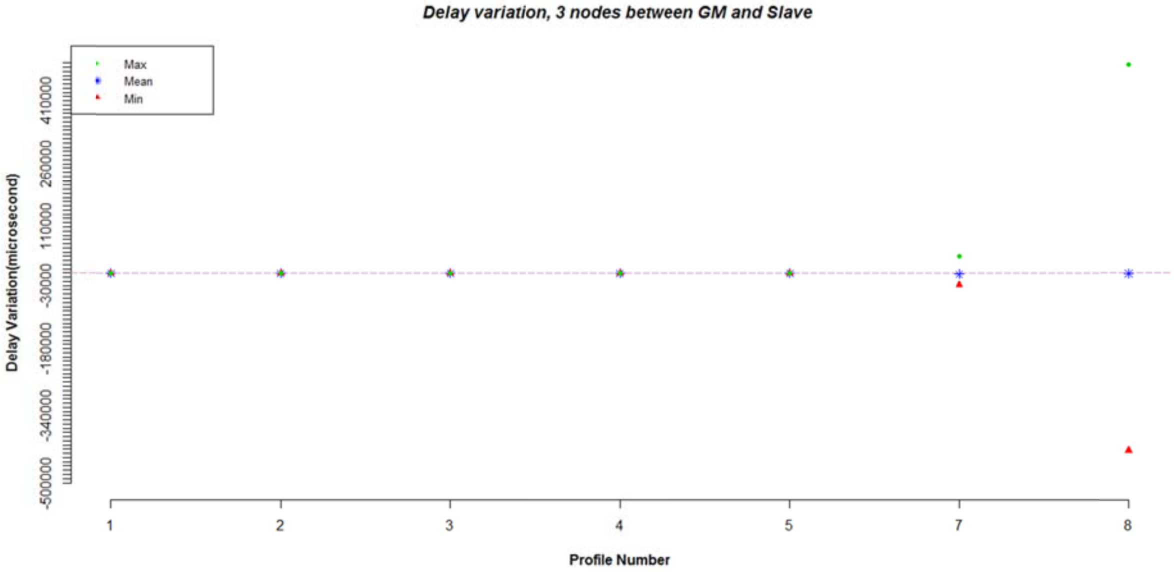

Figure 5-22: **Delay Variation in Different Profile load, three node between GM and Slave**

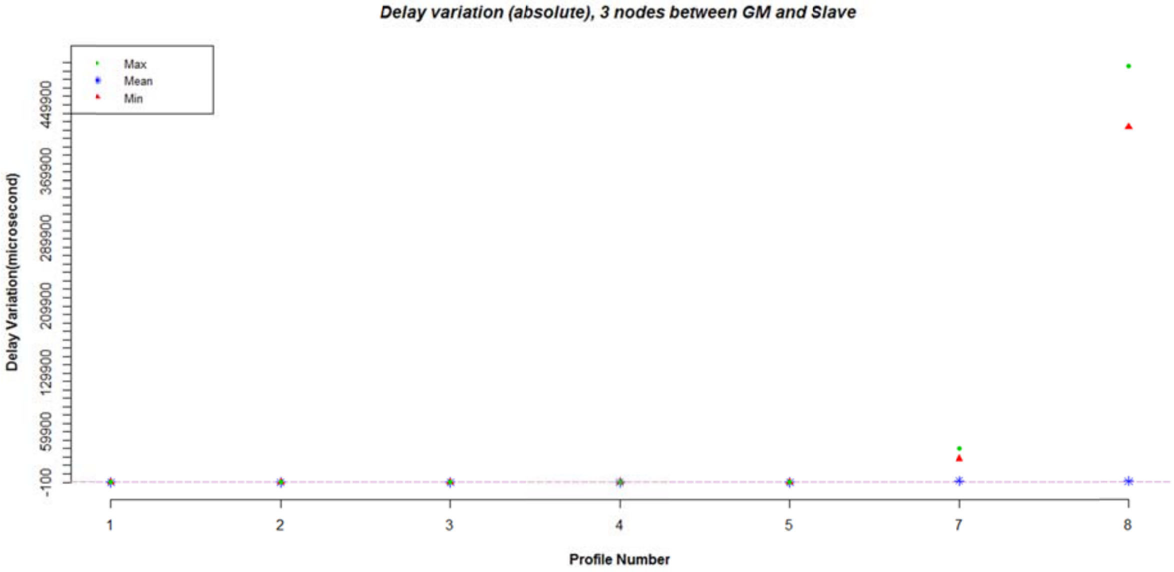

Figure 5-23: **Absolute value of delay variation, three node between GM and Slave**

Table 5-9 shows the standard deviation of delay variation. The high standard deviation of the delay variance in the case of traffic profiles 5 and 8 confirms the anomaly in the delay variance shown in Figure 5-22. In profile 8 GM is subject to a full bandwidth load (100%) of traffic maximum size of frame (1518 byte). Profile 5 contains bursty packets. In this case the middle switches are subject to a bursty load of 50% with 64 byte broadcast frames. This type of traffic will take a lot of input and scheduling processing when these bursts of traffic appear.

Table 5-9: Standard deviation of delay variation with different load profiles, Scenario 3

	Profile						
	1	2	3	4	5	7	8
Std Dev (µs)	21.40	9.205	0.032	19.540	198500	7875	207000

5.2.2.6 Scenario 4

Figure 5-24 shows the delay variation with different traffic load when there are four nodes between GM and slave and Figure 5-25 shows the absolute value of delay variation in this scenario.

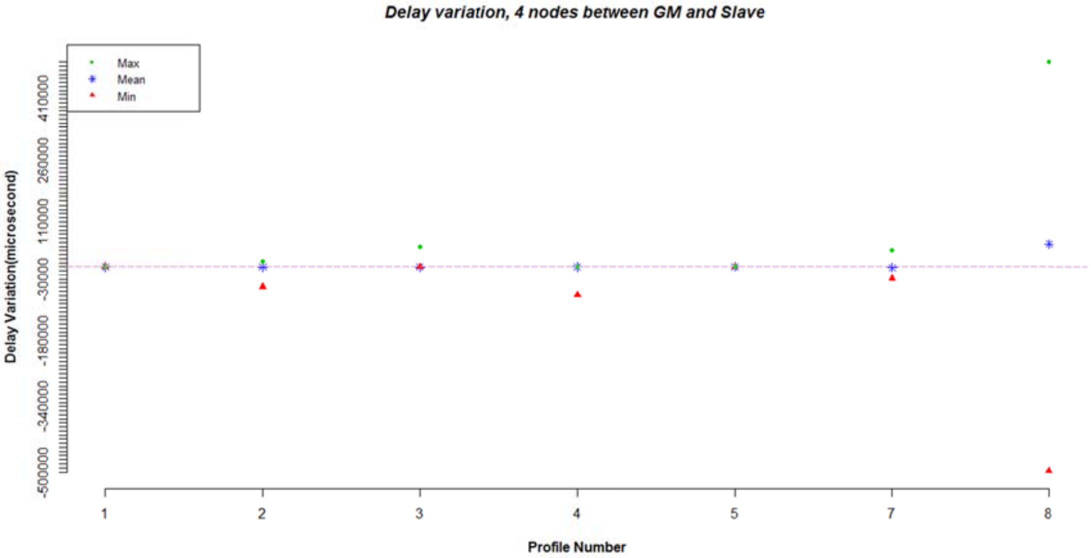

Figure 5-24: Delay Variation in Different Profile load, four node between GM and Slave

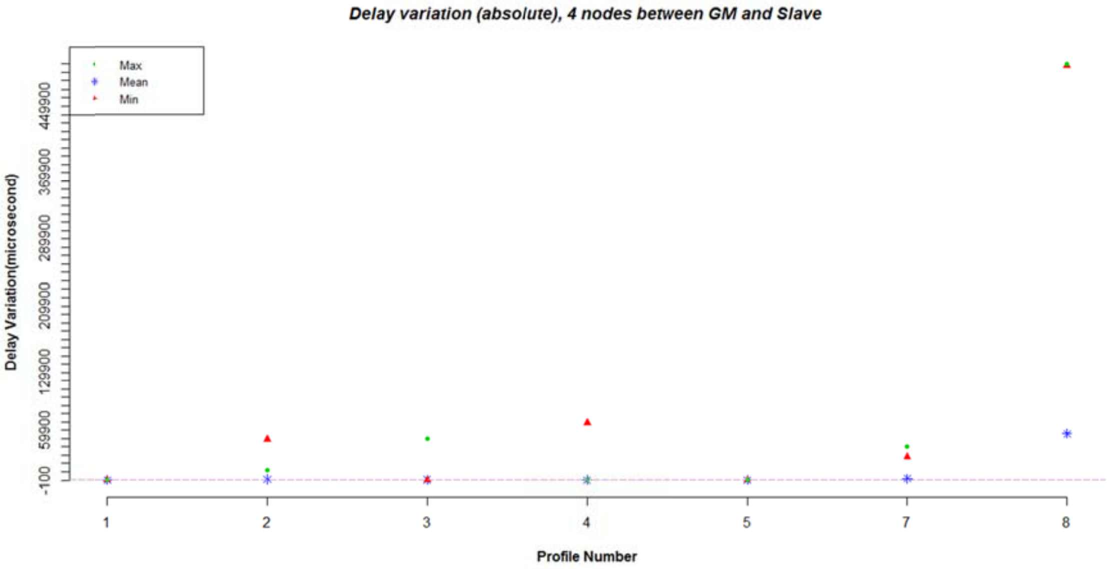

Figure 5-25: Absolute value of delay variation , four nodes between GM and slave

Table 5-10 shows the standard deviation of delay variation. The high standard deviation of the delay variance in the case of traffic profile 8 confirms the anomaly in the delay variance shown in Figure 5-25. Note that in this case the GM is subject to a full bandwidth load (100%) of traffic maximum size of frame (1518 byte). As it mentioned in section 5.2.1.2, in this case many of PTP packets get lost which cause huge delay between GM and slave.

The other anomaly that can be seen in the Figure 5-25 is in profile 2, 3 and 7. In Profile 7 middle switches subject to a very bursty load of 80% with 64 byte broadcast frames. This type of traffic will take a lot of input and scheduling processing when these burst of traffic appear, as contrasted with profile 5 which is just as burst – but due to the larger frame size the time to transmit each frame is longer, hence the delay variance for PTP packets traversing the GM will be lower. The anomaly that is seen in profile 2 and 3 is assumed to occur because of abnormal data. The test should be run additional times in order to increase our confidence in the measurements for these two profiles.

Table 5-10: Standard deviation of delay variation with different load profiles, Scenario 4

	Profile						
	1	2	3	4	5	7	8
Std Dev (µs)	88.845	17862	76611	93.9	211.47	7044	169800

5.2.2.7 Scenario 5

Figure 5-26 shows the delay variation with different traffic load when there are five nodes between GM and Slave. The mean delay variance in different load profiles while the network having continues traffic with 64Byte frames are between 0 and 30 µs which shows an acceptable synchronization. But the delay variation increases when traffic is bursty or has a large frame size.

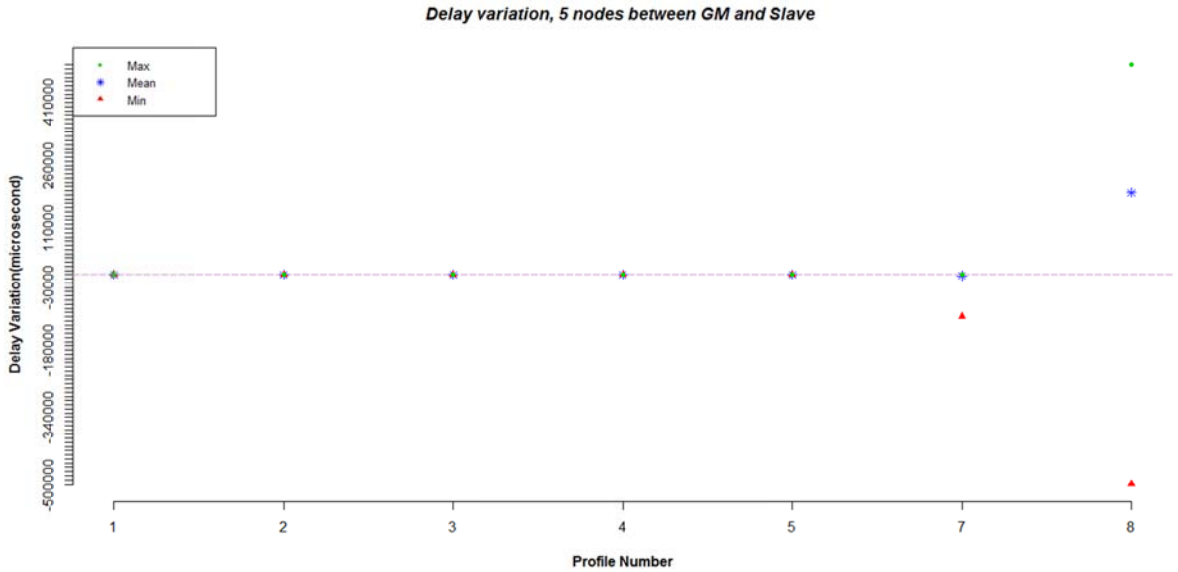

Figure 5-26: Delay Variation in Different Profile load, five node between GM and slave

Figure 5-26 shows the absolute value of delay variation in different traffic load when there are five nodes between GM and slave. Table 5-11 shows the standard deviation of delay variation. The high standard deviation of the delay variance in the case of traffic profile 8 confirms the anomaly in the delay variance shown in. Note that case the GM is subject to a full bandwidth load (100%) of traffic maximum size of frame (1518 byte). As it mentioned in section 5.2.1.2, in this case many of PTP packets get lost which cause huge delay between GM and slave.

The other anomaly that can be seen in the Figure 5-26 and Figure 5-27 is in profile 7. In this case the middle switches subject to a very bursty load of 80% with maximum size of frame (1518 byte). This type of traffic will take a lot of input and scheduling processing when these burst of traffic appear.

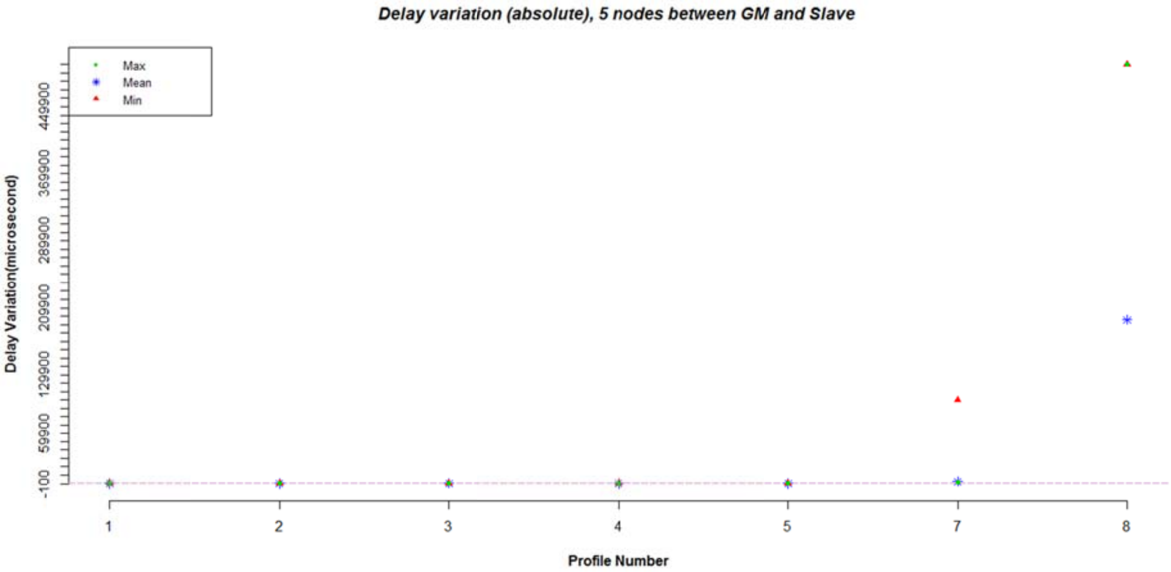

Figure 5-27: **Absolute value of delay variation, five nodes between GM and slave**

Table 5-11: **Standard deviation of delay variation with different load profiles, Scenario 5**

	Profile						
	1	2	3	4	5	7	8
Std Dev (µs)	1.833	0.218	0.435	2.633	4.516	7944	269700

5.2.2.8 Scenario 6

Figure 5-28 shows the delay variation and Figure 5-29 shows the absolute value of delay variation with different traffic load when there are six nodes between GM and slave. The mean delay variance for all loads are in milliseconds which is not acceptable synchronization.

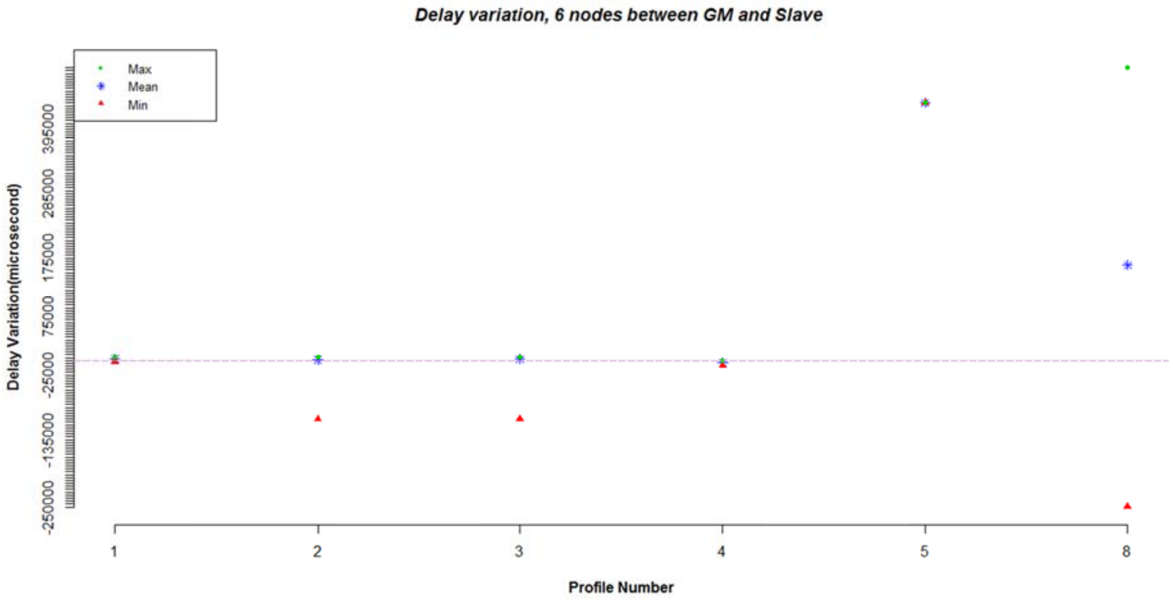

Figure 5-28: Delay Variation in Different Profile load, six node between GM and slave

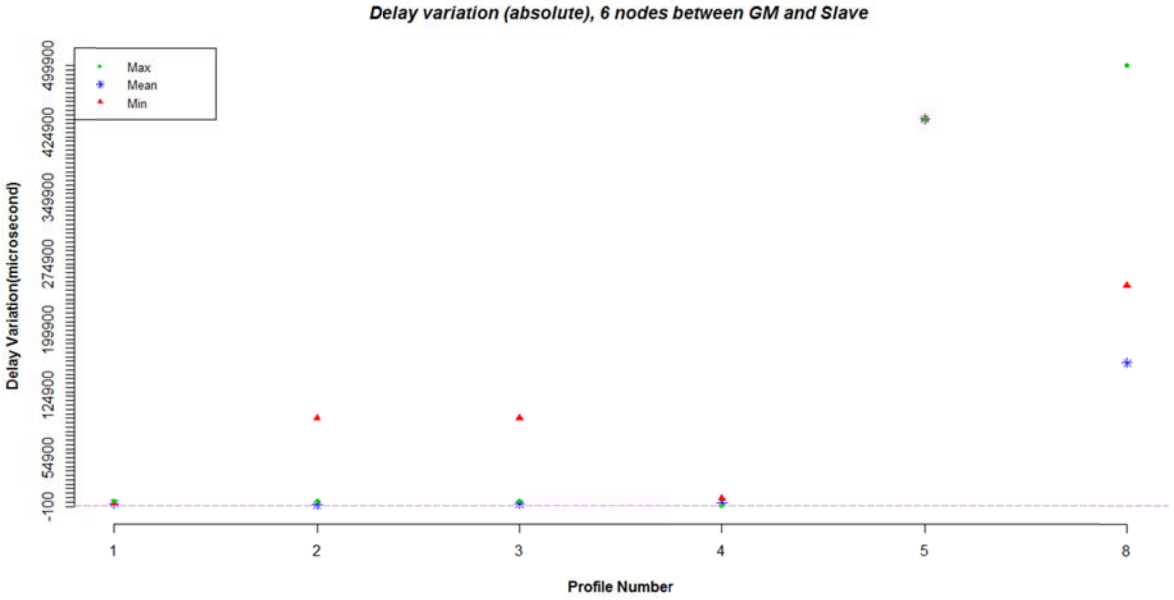

Figure 5-29: Absolute value of delay variation – six nodes between GM and slave

Table 5-12 shows the standard variation of delay variation. Rather high delay variation can be observed in different profiles while in profile 8 the standard deviation of the delay variation is about 243 milliseconds which is a high value with respect to our requirements for synchronization. In Profile 8 the GM is subject to a full bandwidth load (100%) of traffic with maximum size frames (1518 byte). Profile 5 also has a high delay variation. This should be expected considering that this profile contains 50% burst traffic. This type of traffic takes a lot of input and scheduling processing when these bursts of traffic appear. We assume that, by increasing the number of nodes, even a moderate load of burst cannot be handled, while in previous scenarios with a fewer nodes 50% burst traffic in the network would not prevent the slave from being synchronized with the GM.

Table 5-12: Standard deviation of delay variation with different load profiles, Scenario 6

	Profile					
	1	2	3	4	5	8
Std Dev (µs)	1279	9046	10320	2668	210000	243560

5.2.2.9 Scenario 7

Figure 5-30 shows the delay variation and Figure 5-31 shows the absolute value of delay variation with different traffic load when there are seven nodes between GM and slave. As in scenario 6 all the delay variation in different load profiles are in the order of milliseconds.

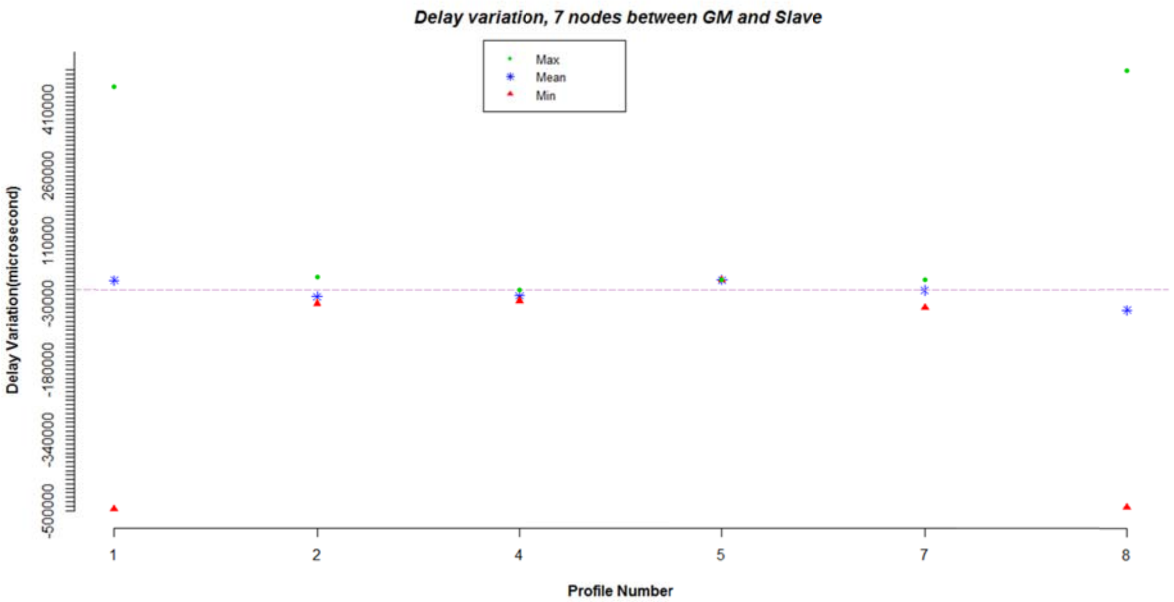

Figure 5-30: Delay Variation in Different Profile load, seven node between GM and slave

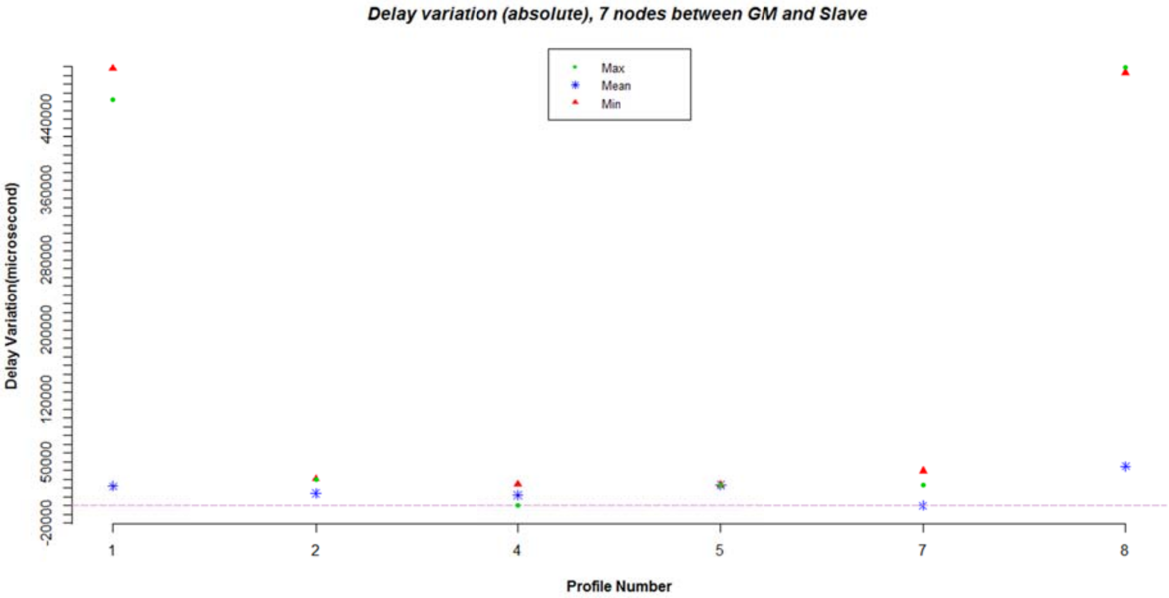

Figure 5-31: Absolute value of delay variation, seven nodes between GM and slave

Table 5-13 shows the standard variation of delay variation. The high standard deviation of the delay variance in the case of traffic profile 1 and 8 confirms the anomaly in the delay variance shown in Figure 5-30.. In profile 8, GM is subject to a full bandwidth load (100%) of traffic maximum size frames (1518 byte). As it mentioned in section 5.2.1.2, in this case many of PTP packets get lost which cause huge delay between GM and slave.

The anomaly in delay variation in profile 1 can be as a result of unnormalized data. In this scenario each test was run only once.

Table 5-13: Standard deviation of delay variation with different load profiles, Scenario 7

	Profile					
	1	2	4	5	7	8
Std Dev (µs)	165460	10373	7031	6758	6777	253760

5.2.2.10 Scenario 8

Figure 5-32 shows the delay variation and Figure 5-33 shows the absolute value of delay variation with different traffic load when there is ten nodes between GM and. All mean delay variance in different load profiles are in the range of milliseconds which shows that the GM and slave are not synchronized.

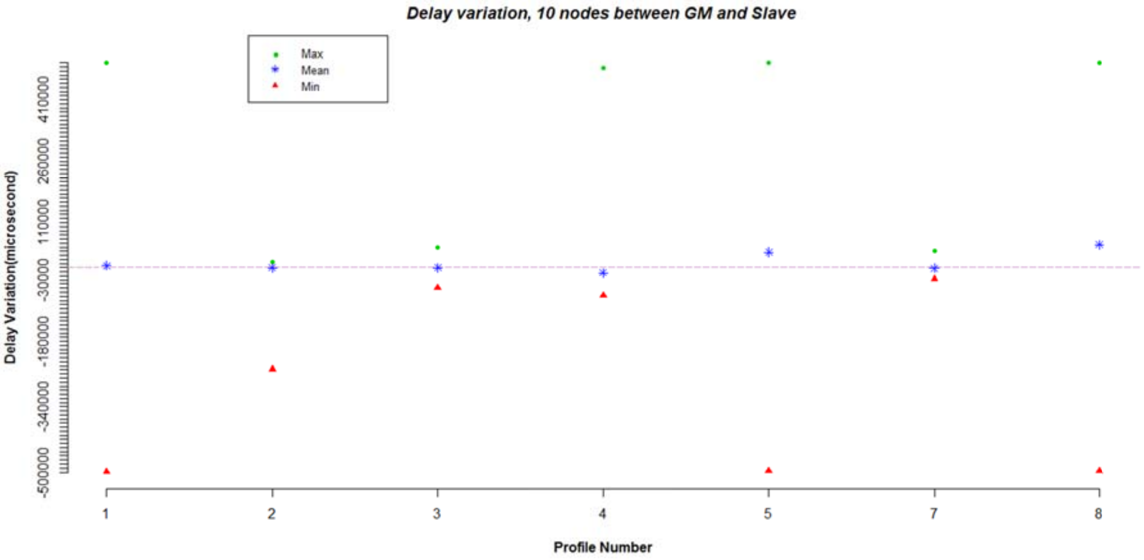

Figure 5-32: Delay Variation in Different Profile load, ten nodes between GM and Slave

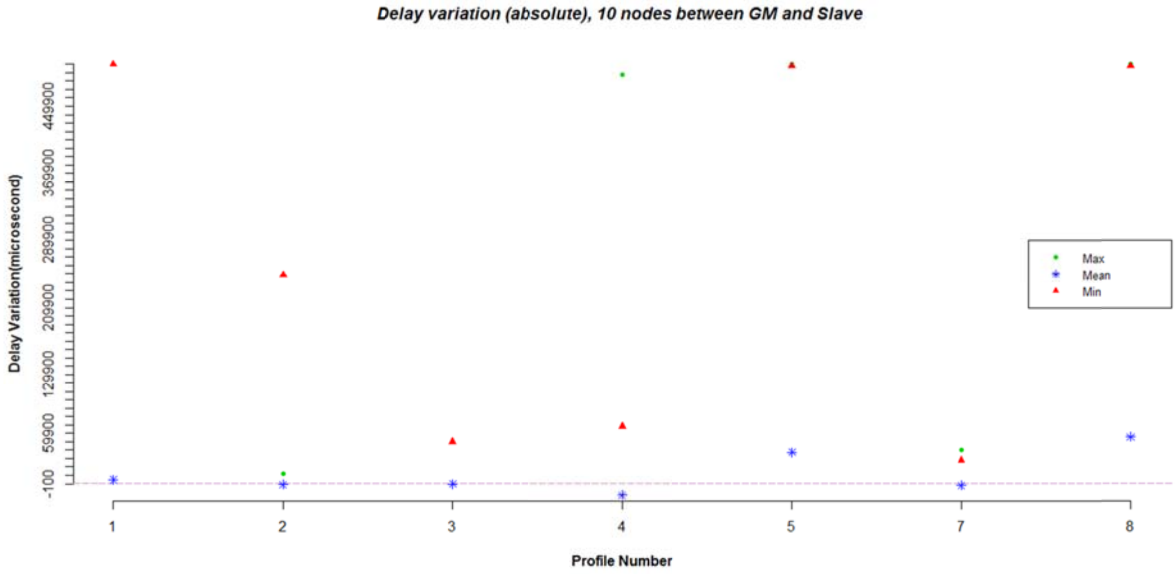

Figure 5-33: Absolute value of Delay Variation in Different Profile load, ten nodes between GM and Slave

Table 5-14 shows the standard variation of delay variation. The high standard deviation of the delay variance in the case of traffic profile 8 confirms the anomaly in the delay variance shown in Figure 5-32. Note that in this case the GM is subject to a full bandwidth load (100%) of traffic maximum size of frame (1518 byte). Our assumption is based on what mentioned in section 5.2.1.2, in this case many of PTP packets get lost which cause huge delay between GM and slave.

Table 5-14: Standard deviation of delay variation with different load profiles, Scenario 10

	Profile						
	1	2	3	4	5	7	8
Std Dev (µs)	96767	8456	76651	9765	211987	147707	253760

5.3 Summary

Table 5-15 shows a summary of the experiments reported in this thesis. A green check mark (√) in the table shows those scenarios with a special profile will be synchronized, while a red cross (×) indicates those scenarios that cannot meet the synchronization requirements.

Table 5-15: Synchronizarion result for different number of nodes between GM and Slave in different load profile

		Profile						
Nodes	**No Traffic**	**1**	**2**	**3**	**4**	**5**	**7**	**8**
0	√	√	√	√	√	√	√	√
1	√	√	√	√	√	√	×	×
2	√	√	√	√	√	√	×	×
3	√	√	√	√	√	√	×	×
4	√	√	×	×	√	√	×	×
5	√	√	√	√	√	√	×	×
6	√	×	×	×	×	×	×	×
7	√	×	×	×	×	×	×	×
10	√	×	×	×	×	×	×	×

6 Conclusions and Future work

This chapter describes the conclusion of this master thesis project and also provides suggestions for future works in this area. In section 6.1, the conclusion of this thesis will be discussed. In section 6.2, the next steps to extend this thesis project will be mentioned. Finally, section 6.3 discusses some reflections regarding the economic, ethical, and environmental aspects of this thesis project.

6.1 Conclusions

This thesis project showed that the PTP protocol can be extend to a MAN by restricting the number of nodes along the path from the GM to the slave node. Overall, we showed that the PTP protocol is deployable in a small MAN with a maximum **five** nodes between master and slave in the case where there is no primary reference clock between GM and Slave.

The limitations in this thesis project stated in section 1.3 prevented us from covering all recent research required for PTP synchronization in packet networks such as primary reference clock (PRC) and also research on other important time aware devices such as BC and TC.

6.2 Future work

As described in section 4.3.9, we did not have a PRC connected to the GM; hence the GM's sense of time drifts with respect to UTC over the course of each experiment. Additionally, the oscilloscope was not connected to an external clock; hence its sense of time also drifts with respect to UTC. For these reasons a future experiment should be done to connect the GM to a PRC and to connect the oscilloscope to a reference clock slaved to the PRC. This would enable long term measurements of the slave with respect to this PRC.

As described in section 3.4.4, the internal algorithm used by the slave to synchronize with the GM is not known; hence we do not know how quickly the slave will discipline its local oscillator. An obvious future experiment would be to measure the performance of this algorithm by synchronizing the slave to a GM disciplined by a stratum 1 clock, and then purposely shifting the timing of the PTP messages to see how quickly the slave adapts.

In this thesis project we have emulated a MAN by considering up to 10 switches on the path between the GM and slave. However, the propagation delay along long physical links between these switches has not been considered. Future work should include this delay when making experimental measurements. It should be noted that this delay does not vary, but it will change the relative proportion of the delay which is subject to variance due to delay in the switches and delay due to traffic loads.

As it mentioned in section 1.3 this thesis did not consider the effect of a boundary clock in the network. In future work, the effect of BCs and TCs in the MAN should be studied. Adding a TC in the PTP protocol provides a peer-to-peer delay mechanism that should be considered in future studies.

A complete study should include experimenting with PTP in both simulation and emulation and compare the results to commercial PTP capable devices. As PTP is not yet a mature standard, the devices are expensive. This comparison would help map the simulated/emulated performance of the protocol to the performance real commercial PTP capable devices.

6.3 Required reflections

The result of experimental study done in this thesis project made an **economical** contribution by suggesting that if the number of nodes along the path between the GM and the slave is limited to four, then PTP can be deployed in a Metro Ethernet Network while meeting the synchronization requirements of the various radio base station technologies listed in Table 2-1. PTP is a new technology which requires time aware devices. As a new technology, the price of these devices is still rather high, so a transition from traditional synchronization protocols to PTP will cost both operators and telecommunications vendors (such as Ericsson) a lot of money. However, the measurement results presented in this thesis suggest that there is an opportunity for both operators and communication vendors to reconsider how they realize synchronization of their base stations, especially with respect to new equipment. Additionally, the thesis raises a question of how to adopt PTP for synchronization of existing equipment.

For business and **ethical** reasons, the name of the device vendors of the experiment used as switches in the experiments for this thesis project were not mentioned. The experiments in this thesis were *not* designed to evaluate and compare these switches, but rather the focus was to understand how the introduction of multiple switches along the path from GM to slave affected the ability to synchronize the slave with the GM. However, in order to facilitate other researchers reproducing the same data using the same configuration, the specifications of the devices used in the experiments were mentioned.

One of the important **social** aspects of this thesis is that using PTP rather than GPS eliminates the need to depend upon an GPS receiver which many be subject to intentional interference or jamming of GPS radio signals; nor is the system not vulnerable to spoofing of GPS radio signals. This can provide greater autonomy to the society and avoid dependence upon the operator of the GPS system. The system still needs a PRC in order to remain synchronized with UTC and the PRC may be necessary to prevent unintentional frequency offsets leading to interference with other users of the radio spectrum. However, this PRC could be provided by a physically secured atomic clock (much as telecommunications operators have done for their time division multiplexed networks).

The main **environmental** contribution of this thesis is achieved by changing the synchronization infrastructure from fully or mostly GPS-base to an IP based solution utilizing NTP and PTP. This reduction in the number of GPS in the network directly reduces the number of GPS receivers that are needed, the number of roof top antennas needed, and the amount of cabling from these antennas to the GPS receivers. However, the larger environmental change is due to the ability to more easily deploy low power indoor base stations, hence potentially greatly reducing the amount of electrical power needed and reducing the amount of transmitted radio power.

References

[1] Mark T. Smith, ‘Introduction to Quantitative Research Methods’, Lecture notes for II2202, KTH Royal Institute of Technology, School of Information and Communication Technology, Stockholm, Sweden, 2012.

[2] James Aweya, Saleh AI-Araji, and Basim Majeed, ‘Timing and Synchronization of Wireless Base Stations over Packet Networks’, presented at IEEE 54th International Midwest Symposium onCircuits and Systems (MWSCAS), 2011, Seul, Korea, pp. 1-4. DOI:10.1109/MWSCAS.2011.6026613.

[3] R. Nath Datta, ‘Understanding the concepts of synchronization and holdover’, EETimes, 03-Oct-2011. [Online]. Available: http://www.eetimes.com/design/communications-design/4213947/Understanding-the-concepts-of-synchronization-and-holdover . [Accessed: 10-Dec-2012]

[4] ‘ETSI TS 145 010 V8.4.0 (2009-06) Technical Specification Digital cellular telecommunications system (Phase 2+)’. Radio subsystem synchronization (3GPP TS 45.010 version 8.4.0 Release 8).

[5] ‘ETSI TS 125 104 V8.7.0 (2009-07) Technical Specification Universal Mobile Telecommunications System (UMTS)’. Base Station (BS)radio transmission and reception (FDD) (3GPP TS 25.104 version8.7.0 Release 8).

[6] ‘ETSI TS 125 105 V8.4.0 (2009-07) Technical Specification Universal Mobile Telecommunications System (UMTS)’. Base Station (BS)radio transmission and reception (TDD) (3GPP TS 25.105 version8.4.0 Release 8).

[7] Hann Kenneth, *LTE Stretches Synchronization to New Limits*. USA: Tellabs, 2012 [Online]. Available: http://www.tellabs.com/info/lte/tlab_sync_wp.pdf

[8] Angela Zhou and Xiaodong Duan, ‘Requirements and viewpoints for backhaul synchronization’, 71st IETF meeting, 2008 [Online]. Available: tools.ietf.org/agenda/71/slides/tictoc-6.ppt

[9] Elham Khorami, ‘Providing accurate time information to a radio base station via a GPS receiver emulator’, Master's thesis, KTH Royal Institute of Technology, School of Information and Communication Technology, Communication Systems, Sweden,TRITA-ICT-EX-2013:1, 2013 [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-118070

[10] ‘Deployment of Precision Time Protocol for Synchronization of GSM and UMTS Basestations’, Symmetricom Inc., San Jose, California, USA, May 2008 [Online]. Available: http://www.syncworks.com/pdf/symmetricom-timeprovider-500/Deployment_of_Precision_Time_Protocol_GSM_UMTS.pdf

[11] K. M. Pesyna, K. D. Wesson, R. W. Heath, and T. E. Humphreys, ‘Extending the reach of GPS-assisted femtocell synchronization and localization through tightly-coupled opportunistic navigation’, in *GLOBECOM Workshops (GC Wkshps), 2011 IEEE*, 2011, pp. 242–247 [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6162445 . [Accessed: 18-Nov-2013]

[12] S. R. Saunders, *Femtocells opportunities and challenges for business and technology*. Chichester, West Sussex, U.K.: Wiley, ISBN: 9780470748169, 2009, Ch. 10, pp. 171. [Online]. Available: http://www.books24x7.com/marc.asp?bookid=33741 . [Accessed: 04-Dec-2013]

[13] ‘General Oscillator Specification : TS05’. Precision Devices INC [Online]. Available: http://proddownloads.vertmarkets.com/download/c81b47b5/c81b47b5-

4721-4ac1-a32d-e6392bca9f1e/original/gs-ts05.pdf

[14] D. Mills, J. Martin, J. Burbank, and W. Kasch, ‘Network Time Protocol Version 4: Protocol and Algorithms Specification’, *Internet Request for Comments*, vol. RFC 5905 (Proposed Standard), Jun. 2010 [Online]. Available: http://www.rfc-editor.org/rfc/rfc5905.txt

[15] D. L. Mills, ‘Network Time Protocol (NTP)’, *Internet Request for Comments*, vol. RFC 958, Sep. 1985 [Online]. Available: http://www.rfc-editor.org/rfc/rfc958.txt

[17] Keith Wing, ‘Difference between NTP and SNTP’, SPECTRACOM, New York, Technical note TN06-101 (C), Sep. 2011 [Online]. Available: http://www.spectracomcorp.com/Support/HowCanWeHelpYou/Library/tabid/59/Default.aspx?EntryId=825&Command=Core_Download

[18] IEEE, ‘1588-2008 - IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems’, IEEE, Standard 1588-2008, Jul. 2008 [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4579760 . [Accessed: 17-Jan-2013]

[19] J. C. Eidson, M. Fischer, and J. White, ‘IEEE-1588 (Trademark) Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems’, DTIC Document, 2002 [Online]. Available: http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA485086 . [Accessed: 30-Nov-2012]

[20] Alexandre van Kempen and Gaël Macé, ‘IEEE1588v1 vs. IEEE1588v2’, SourceForge Ptpd, Documentation for the Ptpd project, Dec. 2009 [Online]. Available: http://sourceforge.net/p/ptpd/code/303/tree/tags/ptpd-2.2.2-rc4/doc/IEEE1588v1_vs_IEEE1588v2.pdf

[21] H. Weibel, ‘High precision clock synchronization according to IEEE 1588 implementation and performance issues’, *Proc. Embedded World 2005*, 2005 [Online]. Available: http://www.engineering.zhaw.ch/fileadmin/user_upload/engineering/_Institute_und_Zentren/INES/Documents/IEE1588/embedded_World_05_Contribution_final_02.pdf . [Accessed: 30-Nov-2012]

[22] R. Cohen, ‘Precision Time Protocol (aka IEEE1588)’, Prague, 2007.

[24] Dirk Mohl, ‘IEEE1588: Precise time synch for realtime automation applications’, *Industrial Ethernet Book*. [Online]. Available: http://www.iebmedia.com/ethernet.php?id=5867&parentid=74&themeid=255&hft=47&showdetail=true&bb=1&PHPSESSID=15n07k02884g8u4i38udkd0cg0

[25] Alexander E Tan, ‘IEEE 1588 and the need for Ethernet Clock Synchronisation’, *Industrial Ethernet Book*. [Online]. Available: http://www.iebmedia.com/index.php?id=5543&parentid=63&themeid=255&showdetail=true

[26] ‘IEEE1588 - Functional Principle Of PTP’. [Online]. Available: http://www.ieee1588.com/IEEE1588_Functional_Principle_of_PTP.html. [Accessed: 12-Nov-2012]

[27] C. Gordon, *White Paper Introduction to IEEE 1588 & Transparent Clocks*. Tekron, 2009 [Online]. Available: http://www.tekroninternational.com/userfiles/file/transparent_clock_whitepaper.pdf . [Accessed: 30-Nov-2012]

[28] Sabina Piyevsky and Gregory Wilcox, ‘How to Implement Time Synchronization on EtherNet/IP’. [Online]. Available: http://fr.rockwellautomation.com/applications/gs/emea/gsfr.nsf/pages/Time_Synchro

nization . [Accessed: 05-Dec-2012]

[29] Jon Beckwith, Harrison de Bree, Jeff Laird, and Ryan Zarick, 'IEEE 1588 Consortium Best Master Clock Algorithm Test Suite Version 0.3', University of New Hampshire, USA, Tech. Doc., Jul. 2011 [Online]. Available: ftp://ftp.iol.unh.edu/pub/1588/BMCA_Test_Suite_v0.3.pdf

[30] HUAWEI, 'Synchronous Ethernet and IEEE 1588v2 Technology White Paper', no. 01, Oct. 2012.

[31] Javvin Press, *Network Protocols Handbook*. Javvin Technologies, 2007.

[32] ITU Telecommunication Standardization, 'ITU-T Recommendation G.810 Definition and terminology for synchronization networks', *Message Sequence Charts (MSC96)*, 1996.

[33] T. Laine, 'New synchronization metrics for packet networks', Aalto University, Finland, 2010 [Online]. Available: https://aaltodoc.aalto.fi/handle/123456789/3316 . [Accessed: 09-Nov-2013]

[34] Paul Lesage and Theophane Ayi, 'Charactrization of Frequency Stability: Analysis of the Modified Allan Variance and Properties of Its Estimate', presented at the IEEE Transactions on Instrumentation and Measurement, 1984, vol. IM.33.

[35] J. Teener and G. M. Garner, 'Overview and timing performance of IEEE 802.1 AS', in *2008 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication*, 2008, pp. 49–53 [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4659212 . [Accessed: 30-Nov-2012]

[36] A. Novick, M. Weiss, K. Lee, and D. Sutton, 'Examination of time and frequency control across wide area networks using IEEE-1588v2 unicast transmissions', presented at the 2011 Joint Conference of the IEEE International Frequency Control and the European Frequency and Time Forum (FCS), San Francisco, CA, 2011, pp. 1–6 [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5977852 . [Accessed: 17-Jan-2013]

[37] Waheed Ahmed, 'Clock Synchronization: Combining IEEE 1588 and Adaptive Oscillator Correction Method', Master's thesis, Carleton University, Department of Systems and Computer Engineering, Ottawa, Ontario, Canada, 2012 [Online]. Available: http://kunz-pc.sce.carleton.ca/thesis/WaheedThesis.pdf

[38] J. Steinhauser, 'A PTP Implementation in OMNET+' [Online]. Available: http://www.vmars.tuwien.ac.at/documents/intern/3019/bakkarbeit_0826623.pdf . [Accessed: 10-Jul-2013]

[39] Y. Liu and C. Yang, 'OMNeT++ based modeling and simulation of the IEEE 1588 PTP clock', in *Electrical and Control Engineering (ICECE), 2011 International Conference on*, 2011, pp. 4602–4605 [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6057471 . [Accessed: 10-Jul-2013]

[40] G. Giorgi and C. Narduzzi, 'Modeling and simulation analysis of PTP clock servo', in *Precision Clock Synchronization for Measurement, Control and Communication, 2007. ISPCS 2007. IEEE International Symposium on*, 2007, pp. 155–161 [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4383791 . [Accessed: 10-Jul-2013]

[41] R. Chen, Y. Zhang, C. Cao, Y. Zhao, B. Li, J. Zhang, and W. Gu, 'Clock synchronization in T-MPLS network via PTP (IEEE 1588 V2)</title>', 2009, pp. 763308–763308–8 [Online]. Available: http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=755238 .

[Accessed: 13-Jul-2013]

[42] P. V. Estrela and L. Bonebakker, ‘Challenges deploying PTPv2 in a global financial company’, in *Precision Clock Synchronization for Measurement Control and Communication (ISPCS), 2012 International IEEE Symposium on*, 2012, pp. 1–6 [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6336634 . [Accessed: 21-Oct-2013]

[43] I. T. S. Sector and O. ITU, ‘Timing and synchronization aspects in packet networks’, *ITU-T Recommendation G-G.8261/Y.1361*, vol. 1080, 2008.

[44] ‘The Network Simulator ns-2: Porting ns-1 functionality to ns-2’. [Online]. Available: http://www.isi.edu/nsnam/ns/ns-1-2.html . [Accessed: 31-May-2013]

[45] Kaiyu Hang, ‘Investigation of Impact of Network Characteristics on QoS in Mobile Networks Student’, KTH, School of Information and Communication Technology, Communication Systems, Sweden , Stockholm, 2012.

[46] ‘Networked Wireless Systems’, *EE384E, CS244E - Networked Wireless Systems*, Winter-2011. [Online]. Available: http://www.stanford.edu/class/ee384e/cgi-bin/references.php . [Accessed: 11-Jul-2013]

[47] Mustafa Ergen, ‘OPNET’, University of California Berkeley, 2003.

[48] OPNET Company, ‘OPNET Modeler Documentation, release 17.1’. [Online]. Available: file:///C:/OPNET/17.1.A/doc/modeler/wwhelp/wwhimpl/js/html/wwhelp.htm#href=Getting%20Started/release_notes_110_overview_modeler.html . [Accessed: 14-Jul-2013]

[49] ‘PTPd - Precision Time Protocol daemon’. [Online]. Available: http://ptpd.sourceforge.net/ . [Accessed: 23-Sep-2013]

[50] T. Kovácsházy and B. Ferencz, ‘Performance evaluation of PTPd, a IEEE 1588 implementation, on the x86 Linux platform for typical application scenarios’, in *Instrumentation and Measurement Technology Conference (I2MTC), 2012 IEEE International*, 2012, pp. 2548–2552 [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6229387 . [Accessed: 24-Sep-2013]

[51] K. Correll, N. Barendt, and M. Branicky, ‘Design considerations for software only implementations of the IEEE 1588 precision time protocol’, in *Conference on IEEE*, 2005, vol. 1588, pp. 11–15 [Online]. Available: http://www.vtinstruments.com/CATALOG/Technotes/Design_Considerations_for_Software_Only_Implementations.pdf . [Accessed: 24-Sep-2013]

[52] Richard Cochran, ‘The Linux PTP Project’. Sourceforge, 13-Dec-2012 [Online]. Available: https://sourceforge.net/projects/linuxptp/files/?source=navbar

[53] ‘P2020E/P2010E Reference Design Board’. FreeScale [Online]. Available: http://cache.freescale.com/files/netcomm/doc/fact_sheet/P2020RDB.pdf?fsrch=1&sr=2

[54] ‘DFE-580TX 4-port Fast Ethernet Server Card’. D-Link, May-2002 [Online]. Available: http://www.dlink.com/-/media/Business_Products/DFE/DFE%20580TX/Datasheet/DFE580TX_Datasheet_EN_UK.pdf

[55] A. Depari, P. Ferrari, A. Flammini, D. Marioli, and A. Taroni, ‘Evaluation of Timing Characteristics of Industrial Ethernet Networks Synchronized by means of IEEE 1588’, presented at the IEEE Instrumentation and Measurement Technology Conference Proceedings, 2007. IMTC 2007, 2007, pp. 1–5 [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4258082 . [Accessed: 24-Jan-2013]

[56] ‘Agilent InfiniiVision 5000/6000/7000 Series Oscilloscopes’. Agilent Technologies, Jun-2009.

[57] Rusty Russell, Alexey N. Kuznetsov, ‘Multicast routing’, *Linux Advanced Routing & Traffic Control HOWTO*. [Online]. Available: http://tldp.org/HOWTO/Adv-Routing-HOWTO/lartc.multicast.html

[58] Joachim Nilsson, ‘Multicast HowT’. 09-Jul-2013 [Online]. Available: http://troglobit.com/blog/2013/07/09/multicast-howto/

[59] ‘IEEE 1588 Precision Time Protocol – Frequency Synchronization Over Packet Networks’. Symmetricom Inc., 20-Feb-2008 [Online]. Available: http://www.scientific-devices.com.au/pdfs/WeTransfer-NZvJB6Cw/Telecommunication%20Synchronization/IEEE%201588%20Precision%20Time%20Protocol%20Frequency%20Synchronization%20Over%20Packet%20Networks.pdf

[60] ‘xStack™ DGS-3324SR’. D-Link [Online]. Available: ftp://www.dlinkla.com/pub/drivers/DGS-3324SR/DATASHEET_DGS-3324SR.pdf

[61] ‘HP-Procurve-2600-series’. HP.

[62] ‘GS748T ProSafe™ Switch 48-port Gigabit Smart Switch’. NETGEAR [Online]. Available: http://www.downloads.netgear.com/files/GDC/GS748Tv1/GS748T_ds_03Feb05.pdf

Appendix A

This appendix will explain the specification of switch that used in this thesis experiment. The order of explanation is based on alphabetic order.

D-Link

Table shows the specification of D-Link switches that used in the experimental study of this thesis [60].

Table 6-1: D-link switch specification

Vendor	**Model**	**Ports**	**Network layer**	**OS Port Standard**	**Function Support**
D-Link	3324SR	-24 auto-sensing -10/100/1000BASE-T ports	Gigabyte Stackable Layer 3 switch	- IEEE802.3 10BASE-T - 802.3u 100BASE-Tx - 802.3ab 1000BASE-T	- ANSI/IEEE 802.3 NWay auto-negotiation - IEEE 802.3x flow control - Auto MDI/MDIX - Port mirroring

HP

Table shows the specification of HP-Province switch that used in the experimental study of this thesis [61].

Table 6-2: HP switch specification

Vendor	Model	Ports	Network layer	Port Standard	Function Support
HP	Province 2626	-24 auto-sensing 10/100 ports; -2dual-personality ports; each port can be -used as either an RJ-45 10/100/1000 port	b Layer 2 and Layer 3 lite feature set	- IEEE802.3 10BASE-T - 802.3u 100BASE-Tx - 802.3ab 1000BASE-T IEEE 802.3ab 1000Base-T Gigabit Ethernet	- IEEE 802.1AB Link Layer Discovery - Protocol (LLDP) - RFC 2819 Four groups of RMON: 1 - (statistics), 2 (history), 3 (alarm) and 9 - (events) - RFC 3164 BSD syslog Protocol - ANSI/TIA-1057 LLDP Media Endpoint - Discovery (LLDP-MED) - SNMPv1/v2c/v3

Netgear

Table shows the specification of Netgear switch that used in the experimental study of this thesis [62].

Table 6-3: NETGEAR switch specification

Vendor	**Model**	**Ports**	**Network layer**	**Port Standard**	**Function Support**
Netgear	GS748T	- 48 RJ-45 connectors for (Auto Uplink on all ports) - 4 SFP slots for fiber Gigabit Ethernet modules	Layer 3-based (DSCP) prioritization	IEEE 802.3 10BASE-T Ethernet - IEEE 802.3u 100BASE-TX Fast Ethernet - IEEE 802.3ab 1000BASE-T Gigabit Ethernet - IEEE 802.3x full-duplex flow control	- IEEE 802.1Q Tag VLAN (up to 64 Static VLAN groups) - Port-based VLAN (up to 48 groups) - IEEE 802.1p (Class of Service) - Port-based QoS (options High/Normal) - Port Trunking - Manual as per IEEE802.3ad Link Aggregation - DHCP client function - Port setting - Web-based configuration, anywhere on the network

Westermo

Table shows the specification of Westermo switch that used in the experimental study of this thesis [58].

Table 6-4: Westermo switch specification

Vendor	Model	Ports	Network layer	Port Standard	Function Support
Westermo	T200 Firmware: 3.89	- 6 RJ-45 Ports - Auto Negotiation Feature Speed - Full and Half Duplex mode - Auto MDI/MDI-X - 4 SFP slots for fiber Gigabit Ethernet modules	Layer2/Layer 3	10/100BASE-TX 100BASE-FX Ports	-GPS synchronised time server -GPS receiver interface supports RS-232 or RS-422 -Satellite signal failure highlighted via SNMP -NTP/SNTP server integrated in switch -Compliant with RFC2030 or T1588

Appendix B

Table 6-5: Mean, maximum and minimum delay variation (in microseconds), GM and slave connected through virtual bridge and facing different amount of delay(in microsecond).

	Profile			
	10	**100**	**200**	**500**
Mean Delay	3.6	47	99	241.8
Minimum Delay	-11	99.8	80	226
Maximum Delay	26.6	122	262	630

Table 6-6: Mean, maximum and minimum delay variation (in microseconds), GM and slave connected through virtual bridge and facing different percentageous of packet loss.

	Profile				
	A	**B**	**C**	**D**	**E**
Mean Delay	-10.8	-349.7	1862	14500	-16750
Minimum Delay	-250	-24950	-10000	-16680	-31600
Maximum Delay	240.5	18000	12600	1000	0

Table 6-7: Mean delay variation (in microseconds) with different numbers of nodes between GM and slave with different load profiles.

	Profile						
Scenario	**1**	**2**	**3**	**4**	**5**	**7**	**8**
0	0.00177	0.00056	-0.0014	0.001	-0.0035	-0.001	-0.0014
1	-14.38	0.458	-1.054	-0.391	-1.569	4.219	6310
2	-6.89	-4025	-0.170	-0.421	-1183	-2070	117780
3	2225	-3765	-0.01	-9.931	-67	-2624	-1249
4	162.6	-966.61	-427.57	-155.86	322.59	-1855	55927
5	-0.35	-0.634	29.365	-1.584	-2.854	-2380	195430
6	2673	1322	2354	-3833	-14449	-	163000
7	22188	-	-14138	-12108	-193.86	-194.51	-44824
8	-10188	-14138	-20238	-25108	-14386	-45310	-44824

.

Table 6-8: **Minimum Delay variation(in microseconds) with different numbers of nodes between GM and slave with different load profiles.**

	Profile						
Scenario	**1**	**2**	**3**	**4**	**5**	**7**	**8**
0	-1	-1	-1	0	-4	-1	-1
1	-200	-9	-5	-3	0.004	-162	-500
2	-82	-20000	-5	-5	-9980	-19000	-500000
3	-10000	-15500	-2	-58.8	500000	-49900	423000
4	-205	-249000	-49800	-320	-205	-28000	-498000
5	-10	-1	-5	-70	-10	-99800	-500000
6	-2000	-100000	-100000	-8540	-250000	-	-250000
7	-498000	-30500	-	-12108	-193.86	-194.51	-44824
8	-49000	-20500	-33650	-45300	-65200	-105600	253760

Table 6-9: **Maximum Delay variation(in microseconds with different numbers of nodes between GM and slave with different load profiles.**

	Profile						
Scenario	**1**	**2**	**3**	**4**	**5**	**7**	**8**
0	0	0	0	1	1	1	0
1	10	15	2	0	0.004	195	7000
2	0	1500	0	0	30	1200	500000
3	475000	1500	0	8.8	500000	15600	496000
4	705	12500	50000	300	-10	40000	500000
5	2	1	1	0	0	15000	500000
6	6000	6000	6000	-10	440000	-	500000
7	463000	-	10373	7031	6758	23500	253760
8	103000	30100	30100	40560	65200	147707	253760

TRITA-ICT-EX-2014:2013:252

www.kth.se