Spring 2011

The Impact of Network Latency on the Synchronization of Real-World IEEE 1588-2008 Devices Using 1588 and non-1588 Aware Switches

Ryan A. Zarick

University of New Hampshire, Durham

Follow this and additional works at: https://scholars.unh.edu/thesis

Recommended Citation

https://scholars.unh.edu/thesis/646

This Thesis is brought to you for free and open access by the Student Scholarship at University of New Hampshire Scholars' Repository. It has been accepted for inclusion in Master's Theses and Capstones by an authorized administrator of University of New Hampshire Scholars' Repository. For more information, please contact nicole.hentz@unh.edu.
The Impact of Network Latency on the Synchronization of Real-World IEEE 1588-2008 Devices Using 1588 and non-1588 Aware Switches

BY

Ryan A. Zarick
B.S. Computer Science, University of New Hampshire (2009)

THESIS

Submitted to the University of New Hampshire in partial fulfillment of the requirements for the degree of Master of Science in Computer Science

May 2011
This thesis has been examined and approved.

Thesis director, Radim Bartoš
Associate Professor of Computer Science

Robert Russell
Associate Professor of Computer Science

Jonathan Beckwith
UNH-IOL 1588 Consortium Manager

5/4/2011
Date
Dedication

To my wife, Cassie Zarick, without her support mentally and financially, there is no way I would be where I am today. To my parents Gregory and Elizabeth Zarick, who taught me the value of hardwork and dedication. To my mentor, Mikkel Hagen, for providing me a Graduate Research Assistantship and mentoring me throughout all of Masters degree.
Acknowledgments

I would like to express my gratitude to my advisor, Radim Bartoš, for his guidance and support throughout my research.

I wish to thank Mikkel Hagen, Jonathan Beckwith, Freescale, IXXAT, Hirschmann, Vitesse, Anue, Analog, Spirent and IXIA for their support of my research.
# Table of Contents

Dedication ................................................................ iv
Acknowledgments .......................................................... v
Abstract .................................................................. xiii

1 Introduction .............................................................. 1

2 Background .............................................................. 4
   2.1 IEEE 1588 ......................................................... 4
   2.2 1588-Aware Bridging ......................................... 9
       2.2.1 Boundary Clocks ...................................... 9
       2.2.2 Transparent Clocks ................................. 10
   2.3 Enterprise Ethernet Switching ............................ 10
   2.4 Cost Considerations ......................................... 11
   2.5 Performance Analysis ...................................... 12
   2.6 Highly Cascaded Transparent Clocks .................. 16
   2.7 Security Considerations ................................... 20
   2.8 Applications .................................................. 21

3 Thesis Statement ..................................................... 23

4 Phase 1: Standard Network ....................................... 24
   4.1 Equipment ..................................................... 24
4.2 Experiment Setup .................................................. 25
4.3 Objectives of the Phase 1 Research ............................. 27
4.4 Phase 1 Experiments ................................................. 28
   4.4.1 Experiment 1: Latency Probability Distribution ............. 28
   4.4.2 Experiment 2: Asymmetric Latency ............................ 30
   4.4.3 Experiment 3: Variance of Gaussian Latency ................. 30
   4.4.4 Experiment 4: The Effect of Congestion on Synchronization . 33

5 Phase 2 ................................................................. 35
   5.1 Experiment 1: The Effect of Congestion on Synchronization with Transparent Clocks and Enterprise Ethernet Switches ............... 36
      5.1.1 Experiment 1: Congested Switch Links via End-to-End Unicast Traffic .......................................................... 38
      5.1.2 Experiment 1: Congested Links via Broadcast Traffic .......... 45
   5.2 Experiment 2: The Effect of Asymmetric Latency on Transparent Clocks and Enterprise Ethernet Switches .......................... 52

6 Conclusion and Future Work ........................................ 61
List of Tables

4.1 Baseline synchronization performance. .......................... 26

5.1 Devices used in Experiment 1. ................................. 38
5.2 Devices used in the asymmetric latency experiments. ....... 54
List of Figures

2-1 How a 1588 device gains synchronization in an existing 1588 network. 6
2-2 How a 1588 clock offset is determined between the Master and the Slave devices [2]. 7

4-1 Topology 1 - 1588 traffic passing through the Anue System Network Emulator. 26
4-2 Topology 2 - 1588 traffic passing through a congested link. 27
4-3 The impact of the four latency probability distributions provided by the Anue System Network Emulator on the 1588 clock synchronization. 29
4-4 The impact of asymmetric latency on the 1588 clock synchronization. 31
4-5 The impact of increasing variance of Gaussian-distributed latency on the precision of the 1588 clock synchronization. 32
4-6 The impact of network congestion caused by background traffic on 1588 clock synchronization. 34

5-1 Generic Topology for Experiment 1 - 1588 traffic passing through the three switches A, B and C. The Meinberg Lantime M600 connected to Switch A will be the master in all of testing for experiments. A Spirent Smartbits 2000 was used to generate broadcast traffic to congest all devices on the networks. 37
5-2 Generic Topology for Experiment 1 - 1588 traffic passing through the three switches A, B and C. The Meinberg Lantime M600 (labeled M) connected to Switch A will be the master in all of testing for experiments. A Spirent Smartbits 2000 was used to generate *unicast traffic* to congest all switches on the networks.

5-3 *Eth-Core Eth-Dom:* Enterprise Ethernet Core and Enterprise Ethernet Dominant Topology for Experiment 1.

5-4 *TC-Core Eth-Dom:* Transparent Clock Core and Enterprise Ethernet Dominant Topology for Experiment 1.

5-5 *Eth-Core TC-Dom:* Enterprise Ethernet Core and Transparent Clock Dominant Topology for Experiment 1.

5-6 *TC-Core TC-Dom:* Transparent Clock Core and Transparent Clock Dominant Topology for Experiment 1.

5-7 Experiment 1- mean synchronization error for 1588 device 2, all nodes on the network were congested with broadcast Ethernet frames.

5-8 Experiment 1- mean synchronization error for 1588 device 3, all nodes on the network were congested with broadcast Ethernet frames.

5-9 Experiment 1- mean synchronization error for 1588 device 4, all nodes on the network were congested with broadcast Ethernet frames.

5-10 *Eth-Core Eth-Dom:* Enterprise Ethernet Core and Enterprise Ethernet Dominant Topology for Experiment 1.

5-11 *TC-Core Eth-Dom:* Transparent Clock Core and Enterprise Ethernet Dominant Topology for Experiment 1.

5-12 *Eth-Core TC-Dom:* Enterprise Ethernet Core and Transparent Clock Dominant Topology for Experiment 1.
<table>
<thead>
<tr>
<th>Page</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5-13</td>
<td>TC-Core TC-Dom: Transparent Clock Core and Transparent Clock Dominant Topology for Experiment 1.</td>
</tr>
<tr>
<td>5-14</td>
<td>Experiment 1 synchronization of 1588 device at location 2 on all topologies in a network congested with broadcast Ethernet frames.</td>
</tr>
<tr>
<td>5-15</td>
<td>Experiment 1 synchronization of 1588 device at location 3 on all topologies in a network congested with broadcast Ethernet frames.</td>
</tr>
<tr>
<td>5-16</td>
<td>Experiment 1 synchronization of 1588 device at location 4 on all topologies in a network congested with broadcast Ethernet frames.</td>
</tr>
<tr>
<td>5-17</td>
<td>Generic Topology for Experiment 2 - 1588 traffic passing through the Anue System Network Emulator. The Meinberg Lantime M600 connected to switch A will be the master in all of testing for experiments.</td>
</tr>
<tr>
<td>5-18</td>
<td>Enterprise Ethernet only Topology for Experiment 2 - 1588 traffic passing through the two enterprise Ethernet switches and the Anue System Network Emulator.</td>
</tr>
<tr>
<td>5-19</td>
<td>Enterprise Ethernet only results for Experiment 2 - The impact of asymmetric latency on the 1588 clock synchronization in an enterprise Ethernet topology.</td>
</tr>
<tr>
<td>5-20</td>
<td>Enterprise Ethernet to TC Topology for Experiment 2 - 1588 traffic passing through an enterprise Ethernet switch, TC and the Anue System Network Emulator.</td>
</tr>
<tr>
<td>5-21</td>
<td>Enterprise Ethernet to TC results for Experiment 2 - The impact of asymmetric latency on the 1588 clock synchronization in a mixed topology.</td>
</tr>
</tbody>
</table>
5-22 TC to Enterprise Ethernet Topology for Experiment 2 - 1588 traffic passing through the a TC, enterprise Ethernet switch and the Anue System Network Emulator.

5-23 TC to Enterprise Ethernet results for Experiment 2 - The impact of asymmetric latency on the 1588 clock synchronization in a mixed topology.

5-24 TC only Topology for Experiment 2 - 1588 traffic passing through the two TCs and the Anue System Network Emulator.

5-25 TC only results for Experiment 2 - The impact of asymmetric latency on the 1588 clock synchronization in a TC only topology.
ABSTRACT

The Impact of Network Latency on the Synchronization of Real-World IEEE 1588-2008 Devices Using 1588 and non-1588 Aware Switches

by

Ryan A. Zarick
University of New Hampshire, May, 2011

Precision Time Protocol (PTP) is a high precision time synchronization protocol designed to operate over a local area network. PTP, often referred to as 1588, is defined by the IEEE Standard 1588™-2008. The protocol theoretically allows synchronization at the nanosecond level. New devices with support for 1588 are emerging into the market, but there have been few studies on real 1588 devices. Our research was broken into two parts: Phase 1 and Phase 2. Phase 1 studied performance of the protocol in an environment where two 1588 devices are connected via a network in which impairments that are typically observed in real networks are introduced and non-1588 devices are present. Measuring the Pulse-Per-Second (PPS) clock outputs of the 1588 boards, we were able to calculate the standard deviation and the mean synchronization error of the 1588 clocks. When we applied latency via network emulators and traffic generators between the 1588 connections, we found that 1588 boards were unable to maintain high accuracy time synchronization under variable and asymmetric latency. The results provide valuable insight into the real-world accuracy and robustness because it is rare that a network will contain neither variable or asymmetric latency. In Phase 2 we studied the impact of latency and high-bandwidth background traffic on 1588 clock synchronization when connected through 1588 and non-1588 aware switches. We found that 1588 aware switches provide higher precision time synchronization in small networks; but in large networks where congestion is present 1588 aware switches were unable to maintain high accuracy clock synchronization without prioritization. Our results also show that having cut-through Enterprise Ethernet switches connected to high
congestion endpoints with priorities enabled is adequate for maintaining sub-microsecond synchronization performance.
Chapter 1

Introduction

There are many applications in need of high precision time synchronization in a closed network environment. In the audio industry, places such as concert venues and theme parks need to have all of their speakers synchronized so that all audio being transmitted is played at the same time. The slightest differences in transmission times can produce discordant sounds which are annoying and disruptive. Manufacturing plants require machines to maintain extremely high levels of time synchronization to produce quality products and lessen the probability of defects. Video production is another example of the need for precise time synchronization. Often times in multiple video stream environments, it is desirable to ensure that videos play at precise times on all of the screens at once. These are a few of the many applications that could benefit from a high precision time synchronization protocol. Expensive proprietary solutions are often used for maintaining high precision time synchronization.

The IEEE 1588\textsuperscript{1} standard defines the Precision Time Protocol (PTP) [2], a high precision time synchronization protocol. Similar to the Network Time Protocol (NTP) [1], the protocol exchanges time-stamped messages to maintain synchronization. While adding timestamps in software is trivial, there is a larger error using

\textsuperscript{1} For the rest of this document we will refer to the IEEE 1588-2008 Standard as “1588”.
this approach to obtain more precise time information, 1588 uses hardware at the lowest level to obtain precise time stamps. This allows devices that implement 1588 to determine the most accurate time a synchronization message was sent or received. 1588 can run over Ethernet but is not restricted to it. There are extensions to the protocol which allow it to run over wireless local area networks and passive optical networks.

In real world networks, latency between devices is a common impairment to overcome. Given that 1588 is a high-precision time synchronization protocol, it is imperative that it maintain synchronization in the most extreme latency environments. There are many different types of latency that could have an effect on the protocol. Asymmetric latency can occur when the transmit side of a copper cable is longer than the receive side. A small length difference could cause a large amount of asymmetric latency. Variable latency can take place when packets take many hops, or pass through intermediary switches, to reach the destination. If any of the hops are congested with varying queue sizes, packets could reach the ultimate destination at different times. The variable latency could have a large effect on the synchronization of the 1588 clocks.

The 1588 standard addresses the latency introduced by many hops between end points by creating 1558-aware switches known as Transparent Clock Switches (TCs). TCs modify fields in the 1588 frames to account for the time the frame spent inside the switch. Transparent Clocks allow for large daisy chains between 1588 devices creating a large fabric.

Companies looking to create a 1588-aware network have many costs to consider in the decision making. A company would have to buy all new 1588 end points for their systems. The company would also have to install brand new 1588 aware bridges such as Transparent Clocks. Since many adopters of the 1588 technology will already have
an existing switching topology in place, it will require a total uproot of their infrastructure. Large enterprises invest heavily in their switches, and replacing these would be out of the question in most cases. To cut costs for adopters of the 1588 technology, our research attempts to determine if a mixed topology of Transparent Clocks and enterprise Ethernet switches can maintain the same high precision synchronization of a Transparent Clock only topology.

The rest of the paper is organized as follows: Section 2 presents the background research of 1588. Section 3 presents the thesis statement. Section 4 explains the experimental results and evaluations of 1588 on standard Ethernet networks when congestion and latency are present. Section 5 explains the experimental results and evaluations of 1588 on Enterprise Ethernet and Transparent Clock switches when congestion and latency are present. Section 6 summarizes our final thoughts on the current study and outlines the ideas for future research.
Chapter 2

Background

2.1 IEEE 1588

IEEE 1588 is the standard defining Precision Time Protocol (PTP), often referred to as 1588. The 1588 protocol is designed to provide time synchronization to the sub-microsecond level of accuracy in a localized network environment. Other protocols existing today, like Network Time Protocol (NTP), do not provide such a high level of accuracy. Unlike 1588, NTP runs over the Internet and therefore requires security overhead and is limited to the connection speeds and latencies existing in the Internet. The GPS protocol provides the same sub-microsecond level accuracy of 1588, but is limited in its application because it requires a view of the sky to gain a connection. The 1588 protocol fits in a niche that would require high precision timing in a localized environment.

The protocol works by having Slave devices synchronizing their internal clocks to a Master device’s internal clock via a Synchronization Mechanism. The IEEE 1588 protocol is a self sufficient one and requires no configuration to attain synchronization. Figure 2-1 shows how a 1588 device connects to an existing 1588 network and gains synchronization. In Figure 2-1 part A, when the 1588 device labeled “Slave” is con-
nected to a 1588 topology it is required to send out multicast ANNOUNCE messages. ANNOUNCE messages are messages that contain information about the device and all of its data sets. In Figure 2-1 part B, all other devices on the network receive the ANNOUNCE messages and transmit their own ANNOUNCE. In Figure 2-1 part C, all devices on the network then perform the Best Master Clock (BMC) algorithm. BMC performs a “pair wise comparison of the data sets” to find the best clock with which to synchronize [2].

In Figure 2-1 part D, once a device decides with which Master to synchronize, the Synchronization Mechanism can begin. The Synchronization Mechanism is a series of four frames transmitted between the Master and the Slave devices as shown in Figure 2-2. The Master device transmits the first frame, the SYNC message, and stores the time the message, was transmitted. Let the time the SYNC message was transmitted from the Master be known as $T_1$.

The Slave device logs the time the SYNC message was received. Let the time the SYNC message was received be known as $T_2$. After the SYNC message is transmitted the Master transmits the FOLLOW_UP message. The FOLLOW_UP message contains the time the SYNC message was transmitted by the Master Device. The Slave then responds with a DELAY_REQ message. The slave stores the time the DELAY_REQ message was transmitted. Let the time the DELAY_REQ message is transmitted be known as $T_3$. When the Master receives a DELAY_REQ message, it responds with a DELAY_RESP message containing the time the DELAY_REQ was received. Let the time the Master receives the DELAY_REQ message be known as $T_4$.

With data points $T_1$, $T_2$, $T_3$, and $T_4$ the Slave can derive the Master to Slave Difference and the Slave to Master Difference. Let the Master to Slave Difference be known as $MDifference$ and the Slave to Master Difference be known as $SDifference$. 
Figure 2-1: How a 1588 device gains synchronization in an existing 1588 network.
Figure 2-2: How a 1588 clock offset is determined between the Master and the Slave devices [2].

The term difference is used to describe the difference in time between the master and the slave through the link. The term Offset in the formulas below is used to describe the time difference between the Master and the Slave. The term Delay is used to describe the time it takes for data to travel between the Master and Slave links. Let the Master to Slave link delay be known as MDelay. Let the Slave to Master link delay be known as SDelay. The formula for the differences can be described as:

\[
\text{MDifference} = \text{Slave Receive Time} - \text{Master Transmit Time}
\]

or

\[
\text{MDifference} = \text{Offset} - \text{MDelay}
\]

and

\[
\text{SDifference} = \text{Slave Receive Time} - \text{Master Transmit Time}
\]
or

\[ SDifference = -Offset + SDelay \]

Inserting the data points the Slave can find that:

\[ MDifference = T_2 - T_1 \]
\[ SDifference = T_4 - T_3 \]

Merging the difference formulas will allow the Slave to solve for the Offset and the Delay.

\[ \text{Offset} = \frac{(MDifference - SDifference) - (MDelay - SDelay)}{2} \]

and

\[ MDifference + SDifference = MDelay + SDelay \]

The Slave assumes the network has symmetric links, meaning the MDelay is equivalent to the SDelay. The formulas can be simplified as:

\[ \text{Offset} = \frac{MDifference - SDifference}{2} \]

and

\[ \text{Delay} = \frac{MDelay + SDelay}{2} \]

Using the data points \( T_1, T_2, T_3, \) and \( T_4 \), the Slave can solve for offset and delay.

\[ \text{Offset} = \frac{(T_2 - T_1) - (T_4 - T_3)}{2} \]

and
\[
\text{Delay} = \frac{(T_2 - T_1) + (T_4 - T_3)}{2}
\]

Having the Offset and the Delay will allow the Slave to accurately adjust it’s time to match the Master in a symmetric network.

### 2.2 1588-Aware Bridging

In large switching topologies there are many daisy chains of switches between any given pair of end points. Since there are many hops for packets to travel through, variable latency will be introduced between each node on the network. Having 1588 nodes on a network where variable latency is present can have a dramatic effect on the clock synchronization of the devices. Frames would arrive at inconsistent times making the time stamps on the frames unusable because of the inconsistency of the link. The 1588 standard addresses the issue of daisy chains and variable latency by creating 1588-aware bridges. There are two main categories of 1588-aware bridges: Boundary Clocks and Transparent Clocks. The difference between a Boundary Clock and a Transparent Clock is that a Boundary Clock does not pass through 1588 traffic while a Transparent Clock modifies 1588 packets when passing through [2].

#### 2.2.1 Boundary Clocks

A Boundary Clock is a 1588-aware switch that does not pass 1588 traffic through to other 1588 devices. Instead the Boundary Clock synchronizes itself to the grandmaster clock and acts as a Master to all other connections on its network. The Boundary Clock segments the network and resolves a 1:N topology, a known issue in 1588 networks [2]. A 1:N topology can happen when all 1588 devices see each other on a network. If this is the case they will all perform the Best Master Clock Algorithm and come up with one Master. This will result in one Master in charge of syncing with every slave on the network. Large 1:N switching topologies could lead to overwhelming
the Master resulting in loss of clock synchronization [2]. Since Boundary Clocks segment the network and will not allow for a master to see slaves beyond the switch this paper will not test environments where Boundary Clocks are present.

2.2.2 Transparent Clocks

A Transparent Clock (TC) is a 1588-aware switch that passes all 1588 traffic through to 1588 devices, and modifies fields inside the packets to compensate for the time the packet was in the switch. Transparent Clocks introduce a 1:N topology and are not suitable for large networks without Boundary Clocks. A well designed 1588 network would have a mixture of Boundary Clocks and Transparent Clocks to break up a 1:N topology and still maintain high clock precision. Our research will use Transparent Clocks and compare them to existing enterprise Ethernet switches to see if an all 1588-aware bridging environment is required to maintain high accuracy clock synchronization [2].

2.3 Enterprise Ethernet Switching

In today’s modern enterprise Ethernet environments, all switches have the ability to prioritize traffic, via VLAN tagging. VLAN tagging allows the ability to create virtual networks, segmented from other traffic on the network. Enterprise switches have the ability to prioritize both VLANs and specific traffic within VLANs via the priority field, which is a 3 bit field containing a packet’s specified priority. Using VLAN tagging in a 1588 network could allow the switches prioritize all 1588 traffic over other traffic on the LAN. This would make the 1588 traffic be in the switches' buffers for significantly less time than all other traffic in the presence of congestion. Our research will look to see if prioritizing 1588 traffic can lead to higher clock precision in the presence of congestion.
2.4 Cost Considerations

There are many cost considerations to take in account when deciding whether or not to implement a 1588 network. M.D.J. Teener et al. presented a low cost implementation of a 1588 TC [13]. They ran two experiments: a master connected directly to a slave and the master and slave connected with one to five TCs in between. Teener shows an appropriate average error within 20ns for up to six Transparent Clocks between the master and the slave. The results show TCs are becoming cheaper and how new adopters of 1588 could implement a large scale network for a low cost. A paper by Han and D.K.Jeong identifies three accuracy issues with 1588: asymmetric propagation delay, quantization errors, and unstable clock sources. He presents a case study of a low cost gigabit TC and how it can achieve sub microsecond accuracy at a relatively low cost [15]. The author has further research underway to address quantization errors by increasing the resolution of timestamps. He expects to achieve accuracy on the order of a few nanoseconds. In a paper by K. Lee and E. Song the authors use the unified modeling language (UML) to model the 1588 standard and produced C++ code based on the model [16]. The model should allow rapid, low cost 1588 implementations. The authors plan to offer a reference design in the future . Unfortunately, not all adopters will want to install all new switches even at a reduced cost. New adopters will have to consider installation, training, and maintenance costs which could out weigh the benefit of cheap TCs. Our research attempts to determine if non-1588 aware switches can provide the same reliability TCs provide.

J.C. Eidson et al. proposed a device that can be retrofitted to existing bridges to provide highly accurate timing using 1588 [14]. The authors came up with the idea of modifying existing bridges because replacing bridges with 1588 aware devices could be very expensive. Spyder, the name of their device, is a network device that intercepts and modifies packets. It adds a proprietary time-stamp to incoming packets,
measures actual residence time in the switch, and removes the proprietary time-stamp and corrects the actual time-stamp on outgoing packets. Spyder increases the synchronization accuracy of a loaded bridge from 70 ms to 80 ns (syntonized spider) - 800 ns (un-syntonized spider). Having to add a Spyder to each port in a switch could be tedious and require additional training for IT departments. Our research uses existing switches without add-on devices.

2.5 Performance Analysis

M.D.J. Teener and G.M. Garner experiment with two 1588 devices with six 1588-aware bridges. An oscilloscope was used to measure time pulses between the two 1588 devices and a traffic generator connected to each 1588-aware bridge to create traffic load [17]. They performed four tests: no traffic from the generators, constant 75% load of 1088 byte frames, random traffic sent at a rate of 75%, and constant 1088 byte frame load of 100% utilization. The authors presented no results, but an amendment to their paper should be available in the near future. These tests were performed using 802.1AS aware devices. 802.1AS is a standard defined by the IEEE that uses 1588 and builds upon it for audio and visual technologies. 802.1AS is a subcomponent of a larger technology known as Audio Visual Bridging (AVB). 802.1AS defines a time synchronization protocol built upon 1588. For the rest of this document we will refer to AVB bridges and end stations as 1588-aware devices.

J. Burch et al. present several TC results under real network conditions. After performing these tests the authors developed a new test methodology called “Correction Factor Error” [18]. Verifying performance of 1588 systems is complicated because of: 1588 device quality, network topology, and network load. Most studies use a test bed with a master/slave and a 1 pulse per second signal hooked up to an oscilloscope. The authors contend this test bed/oscilloscope methodology is inadequate for answer-
ing more challenging questions such as: will a system of Boundary and Transparent Clocks meet end requirements of accuracy? How are Boundary and Transparent Clocks affected by data/control plane traffic load? Do Boundary and Transparent Clocks introduce asymmetry? How does performance change as the number of hops increase? How does performance degrade as the number of Clocks increases? The authors utilize an Agilent N2X system to measure the quality of a TC. The N2X sends many PTP frames through the TC, measures the resident time of the packets and compares its value with the value reported by the TC in the packet’s correction factor (CF) field. By plotting this difference versus injected latency over thousands of packets, the authors were able to identify a very difficult syntonization problem. This testing addresses many issues with current experiments on Transparent Clocks. For our experiments we are looking to compare Transparent Clocks to enterprise Ethernet switches. Since we are not testing the quality of Transparent Clocks we do not use the “Correction Factor Error” methodology.

In another recent study, Takahide Murakami et al. investigated queuing delays in a paper titled “Improvement of Synchronization Accuracy in IEEE 1588 Using a Queuing Estimation Method” [5]. In this study, the researchers attempted to improve the accuracy of synchronization in a 1588 network with a new queuing estimation algorithm. Their new queuing estimation method utilizes probing packets to accurately determine the delays in the network. The researchers analyzed the accuracy of their method via simulation with the commonly used simulator OPNET and showed that their new method was effective in improving synchronization.

P. Moreira et al. used White Rabbit, the name of a project using Ethernet as both a deterministic (sync) data transfer and a timing network [19]. The project provides sub-nanosecond timing accuracy and an upper bound guarantee for high priority messages. The goal is to synchronize up to 1000 nodes with accuracy less
than 1 nanosecond. They were able to achieve sub-nanosecond accuracy by adding a phase measurement module that measures the phase difference between master and slave and adds it to the 1588 correction field. The authors implemented it in Xilinx FPGAs and measured between two nodes. They found ±380 picoseconds, concluding the approach is possible to implement. For our research we do not want to modify the way 1588 does timing. We are looking to find if we can use enterprise Ethernet switches in a mixed topology and still maintain high precision timing.

D. Kohler implements 1588 in an FPGA with multiple ports and a simplified master in another FPGA[22]. With an oscilloscope he measures synchronization via a one pulse per second signal output on the FPGA. The author was able to show the slave synchronization to the master on average within 1 nanosecond. He was also able to show that synchronization intervals from 3 to 2 to 1 second improve accuracy drift jitter from 900 nanoseconds to 180 nanoseconds. In another paper, P. Ferrari et al. ran experiments on a real time 1588 network with Transparent Clocks and 1588 devices [20]. They used a one pulse per second signal to compare synchronization measurements between the 1588 devices on the network. They tested PTP (1588) and Precision Transport Clock Protocol (PTCP) to find synchronization accuracy of their test setup. PTCP is a protocol used between Transparent Clocks. The authors found that PTP (1588) with two second Sync message interval had a standard deviation of 100 nanoseconds. They also found PTCP with a four millisecond synchronization interval had a standard deviation of 73 nanoseconds. The authors concluded PTP and PTCP were adequate for many high precision timing applications. For our research we use the same techniques to find the accuracy of the clock synchronization.

S. Schriegl and J. Jasperneite examine the environmental effects of −40°C to +80°C along with vibrations up to 5 g on 1588 devices [21]. Their goal was to have topologies with at least 100 devices synchronized with an accuracy smaller than
1 microsecond. The authors use two FPGA 1588 boards, with one subjected to environment changes the other not. The authors had two chambers at −35°C and +80°C and moved the FPGA via servo arm between the two rooms. The authors found a frequency variation over the vibration test of below 80 ns. The authors found a frequency variation over the temperature range of 50 ns. The authors were able to show 1588 was able to maintain high clock synchronization accuracy in the most dire circumstances. Our research looks to show how 1588 can withstand environments where asymmetric and variable latency are present.

In “Packet Delay Variation Management” by Dinh Thai Bui et al., the researchers identified how Packet Delay Variation (PDV) within real networks, especially within the telecom industry, greatly affects the performance of the IEEE 1588 protocol [6]. The authors identified several means by which to modify the severity of PDV and finally introduced a modification to the 1588 protocol designed to reduce the PDV. The proposed modification increases the rate of 1588 messages so that the robustness of the protocol is improved in the face of larger PDVs. The authors identified the importance in understanding how variations in latencies and packet delays cause problems in 1588 synchronization. We hope to continue this work and increase the understanding of how latencies affect the protocol. Patrick Loschmidt et al. recently examined the accuracy of hardware time stamping [8]. They produced some interesting results measuring the effect that increasing the synchronization time interval had on the standard deviation between two clocks. Loschmidt et al. were able to show that about 0.5 seconds appeared to be the optimal time interval which produced the most appropriate results.

In a recent Master Thesis presented by Hamza Abubakari, he examined how the unique characteristics of a wireless network make time synchronization challenging [9]. In his thesis, Abubakari identified a common problem in wireless networks of
packet loss and proposed some new techniques to mitigate the loss of packets. Wired networks like we are investigating do not generally lose a large number of packets unless congestion occurs. Thus we are attempting to better understand how packet loss during congestion can affect synchronization.

A paper by Jiho Han and D.K. Jeong inspired our methodology [10]. In their work, Han and Jeong examined 1588 devices connected to a network of Transparent Clock bridges. They also included a traffic generator and measured the clocks on the devices using a sampling oscilloscope. Our Phase 1 research builds upon this by continuing to use the oscilloscope but looking at non-time aware bridges along with utilizing the Anue System Network Emulator to examine latency issues more closely. Our proposed research looks to build upon the Phase 1 research by also using more 1588 aware devices and TC bridges as well as non-1588 aware legacy bridges.

2.6 Highly Cascaded Transparent Clocks

In a recent study by Jeff Burch et al. entitled “Verifying the Performance of Transparent Clocks in PTP Systems” the researchers investigated how Transparent Clock (TC) bridges behave under realistic network conditions [4]. The IEEE 1588-2008 standard defines a Transparent Clock bridge as a bridge that is 1588 time aware and modifies the time stamps of frames as they pass through the network so that timing information is the most accurate. Having 1588 aware switches allows for daisy chains of bridges to be connected without loosing high performance metrics. The TC bridges compensate for variable latency found in large Ethernet networks. The researchers used two released TC bridges and one prototype TC bridge to measure the synchronization accuracy and develop a test technique to measure the accuracy of TC bridges which they called Correction Factor Error. Our Phase 1 research looks to extend this work by examining how the protocol works with legacy and TC bridges
when latency and high traffic throughput are present.

In a paper by C. Na et al. the authors highlight the influence of frequency drift on synchronization [23]. They propose a new algorithm to improve the protocol. The following list are factors affecting quality of 1588 synchronization: stability of oscillators, resolution of message time stamping, frequency of sync messages, and propagation delay variation. The peer-to-peer Transparent Clock implementation of 1588, guarantees a synchronization precision of 1 microsecond for topologies of no more than 30 consecutive slaves. The authors develop an error expression which enhances the performance of 1588 by removing bias in the synchronization error which accounts for frequency drift at the master. The authors prove their algorithm using simulations. The authors attempt to fix a known issue in the 1588 protocol. In our studies we look to find out if 1588 truly requires an all 1588-aware network. We look to determine whether or not a mixed topology can maintain high network synchronization.

1588 relies on two processes: the timing propagation process and the line delay estimation process. In a paper by R.L. Scheiterer et al., the authors claim it is important to improve both so that the limit on the number of slaves can be increased in a 1588 network [24]. An earlier paper proposed an error expression to account for the drift in the master clock due to temperature and accelerations. This paper proposes a further correction for the n\textsuperscript{th} slave in a line to account for drift that happens between the 1\textsuperscript{st} slave and the n\textsuperscript{th}. The authors planned to report results in a future paper. Our research has up to three switches between end points which is well within the existing capacity of a 1588 network. Therefore the algorithm is not used in our research.

Amin Abdul and Petru Lupas in their paper, “Convergence of Frequency, Time and Data over Ethernet Networks” have examined how the IEEE 1588 protocol works
over a typical Ethernet network with other network traffic [7]. They focused on a ring network topology, but many of the concepts still apply to the standard switched Ethernet network we are investigating. They investigated how the 1588 protocol handles standard network start up and basic disruptions. Our research is planning to extend this further by understanding how the 1588 protocol handles different levels of congestion. R.L. Scheiterer et al. test the impact of network latency on 1588 devices when the oscillator of the grand master clocks’s frequency is varied [26]. The authors create a simulator to examine how increasing the frequency will effect a line of 1588 devices with one microsecond accuracy. The authors’ results show the lower the frequency the greater the accuracy of the clocks. These results provide a rough understanding of the effect frequency changes can have on a large 1588 network. The results are not completely accurate because the simulator did not include all of the effects that influence clock synchronization. Fortunately, our research will not have to simulate a large 1588 network, because we have real Transparent clocks and 1588 devices for testing.

In a paper by C. Na et al., the authors look to see how varying temperature in the master of a 1588 network will effect the clock synchronization of the 1588 devices [26]. They found that as the master was heated, the slaves were able to follow the frequency drift of the master unless the master would change drift directions between sync messages. The authors offer several solutions for resolving the drift: minimize temperature effects through insulation, choose oscillators that are temperature independent, shorten synchronization intervals, use QoS to remove bridge delays, or modify the slave error correction to compensate. The authors derived an error formula for calculating the drift. In another paper by R.L. Scheiterer et al., the authors look to determine how the clock synchronization of a 1588 network is effected by the slaves drifting instead of the master in a large topology [27]. The authors explain it is
much less expensive to protect the master’s internal drift by controlling temperature than the slave’s drift. They explain that it is a more realistic scenario where the master will be in a more controlled environment. The authors conclude through testing that the master drift is a order of 10 more important than the slave drifts. The authors were able to conclude that investing in a more expensive master is highly desirable. They also come up with an error formula for slave drift that should be included in the drifting slave and every slave down the line of cascaded slaves. Our research will consist of testing 1588 networks in a normal environment where temperature will not effect the 1588 devices. For our testing we will be using a Meinberg Lantime 600 for our master in all of our experimental setups. The Meinberg is a 1588 device with a GPS specifically designed to be a Grand Master Clock in a 1588 environment. Using a 1588 device designed for being a master in a network will make our results more reliable.

In a paper by S. Meier et al., the authors present an FPGA with three port bridge, a TC, and a ordinary clock with other protocol logic [30]. The goal of their experiments was to provide a robust system in a small package that relies only on a single crystal oscillator. The oscillator can cope with fast temp changes and accelerations. The boards had a temperature range of $-40^\circ C$ to $+80^\circ C$ and can cause drift up to 96 nanoseconds within a 32 millisecond cycle. The boards can also handle a heavy mechanical load (shock/vibration), causing a drift up to 80 nanoseconds within a 32 millisecond cycle. Our research does not require 1588 boards that can handle extreme environments because we are only testing the 1588 protocol under normal environments. We are looking to determine if under normal environments non-1588-aware switches can be used in a mixed fabric and still maintain high precision timing.
2.7 Security Considerations

In a paper by A. Treytl et al., the authors explain how the convenience of synchronized Ethernet opens new vulnerabilities [31]. The main security goals of industrial systems are: integrity, authentication, and availability. The 1588 environments would become susceptible to attacks such as: direct attacks on masters/slaves, byzantine masters ("babbling idiot" master), and disturbance of the control loop. The authors point out countermeasures to these attacks such as: cryptography, authentication and key system, hardware time-stamping, and QOS monitoring to prevent hostile delay injections. In another paper by A. Treytl et al., the authors identify the following threats to 1588 networks: Denial of Service (DoS), byzantine master, interruption of control loop, removal of control packets, packet manipulation, packet insertion and selective packet delay [33]. The standard allows for a security flag to be set in messages and then requires an authentication Type Length Value (TLV) which contains the message authentication code, also called Integrity Check Value (ICV). The standard defines three types of bridges: security-unaware, security-aware and security capable. The authors conclude 1588 security works well, but must be used carefully and in mixed security environments it is especially important to have policies that protect the secure sections of the network. Our research uses security-unaware bridges because the objectives of the testing have no reason to have security enabled. Future testing could be done to compare the performance of security enabled 1588 bridges vs. security disabled 1588 bridges.

In a paper by T. Koskiahde et al., the authors define a sensor network architecture for military and crisis management [32]. The authors describe how using 1588 in a sensory network will allow military systems to achieve high accuracy in large systems. The authors tested 1588 with Ethernet cables up to one kilometer, and showed that the 1588 network was capable of maintaining high accuracy in large fabrics over long
distances. Our research does not test distance or security, and is only looking to see how 1588 will perform in mixed topologies and when asymmetric latency is present.

2.8 Applications

In a paper by A. Vallat and D. Schneuwly, the authors study the statistical properties of asymmetric delay via simulations [34]. The authors assume cables have no asymmetry and most of the asymmetry comes in the queues due to different traffic loads. They ran simulations of up to 6 hops and concluded that 1588 can suffice for telecommunications as long as the network does not approach congestion. In small networks with good QoS, the authors conclude 1588 bridges are not needed. They also conclude if some or all bridges are not 1588 aware, then congestion has a severe effect on clock synchronization. Our research attempts to determine how asymmetric latency and congestion effects clock synchronization with all TC, all enterprise switches, and mixed TC and enterprise switch environments.

In a paper by K. Harris, the author explains how having 1588 in industrial environments will increase the quality and speed by which products can be created, resulting in cheaper production costs [37]. Harris tests how slaves recover from large changes in the master clock. The author concludes 1588 is adequate for industrial uses, but there will be challenges during migration. In another paper by S. Rodriques, the author describes a topology using Synchronous Ethernet (SyncE) and IEEE 1588; SyncE can be used to transmit clock information from core to edge and 1588 can be used from the edge to access layer [35]. Our research only uses the 1588 time synchronization protocol, because we are attempting to determine the environments in which 1588 will thrive.

In a paper by R. Subrahmanyan, the author describes implementation considerations when implementing 1588 in a telecommunication environment [36]. The author
describes how telecommunication environments are different from those of normal
1588 applications because they involve long distances with variable latency. The
author explains how special algorithms will need to be used for filter delay, loop
bandwidth and local clock quality. Our research is attempting to determine how the
existing 1588 protocol performs when congestion and asymmetric latency are present.
Therefore using optimized algorithms for adjusting for delay was not used in our test-
ing:
Chapter 3

Thesis Statement

The goal of this thesis is to build on the previous research outlined above and to investigate the following questions:

- How does using TC bridges in a 1588 network affect the time synchronization of 1588 devices when compared to consumer and enterprise bridges?

- What is the effect of injected latency (both symmetric and asymmetric) on the time synchronization when TC bridges and enterprise bridges are present?

- How does using both TC bridges and enterprise bridges in the same network affect the time synchronization of 1588 devices?
Chapter 4

Phase 1: Standard Network

In our Phase 1 research we studied how 1588 devices would work when connected via legacy Ethernet bridges. The results of this study were published in the ISPCS 2010 Conference titled “The Impact of Network Latency on the Synchronization of Real-World IEEE 1588-2008 Devices” [12]. The following sub-sections describe the experimental setup and the results published in the ISPCS 2010 conference.

4.1 Equipment

To carry out tests we acquired two Analog BF518 EZ-Kit 1588 development boards running the IXXAT 1588 protocol stack\(^1\), two off the shelf consumer grade Netgear 1 Gb/s Switches, a Spirent Smartbits 2000 traffic generator/analyzer chassis with six 1 Gb/s cards, and an Anue System Network Emulator. The Spirent Smartbits is used to generate background traffic to cause network congestion results in increased latency. The Anue System Network Emulator allows us to emulate latency independently in each direction. The network emulator has many capabilities. It can act as an in-line traffic analyzer, allowing it to save traces of all the traffic passing through it.

\(^1\)For the rest of this document this device will be referred to as the IXXAT 1588 board
It also has the ability to inject multiple types of errors including frame corruption, dropped frames, out of order frames and various types of latency impairments. The Netgear switches fairly represent consumer-grade switches that do not support the 1588 protocol. While particular 1588 switches are designed to be highly accurate in their timing and are able to account for the intra-switch delays, the goal of this project is determine how the 1588 protocol performs over non-1588 switches. With these devices we were able to emulate a range of real-world latency scenarios between the two IXXAT 1588 boards. The degree to which the IXXAT 1588 boards are synchronized was measured using a Tektronix oscilloscope attached to the Pulse-Per-Second (PPS) outputs of the IXXAT 1588 boards.

4.2 Experiment Setup

Figures 4-1 and 4-2 outline the setup of the experiments. Table 4.1 shows the baseline synchronization performance of our test setup. Two measures of synchronization provided by the oscilloscope are used throughout this paper: mean synchronization error and standard deviation of the error. The table shows the performance for the following three setups: the boards directly connected with a short crossover cable, the boards connected to the two interconnected switches, and finally, the boards connected to the switches with the network emulator, configured to add no additional impairments, and inserted between the switches.

The first topology (Topology 1 shown in Figure 4-1) involved connecting each of our two identical IXXAT 1588 boards to a Netgear 1 Gb/s switch. The two switches were then connected to the Anue System Network Emulator. The two IXXAT 1588 boards are only capable of a 100 Mb/s rate, while the Anue System Network Emulator only supports a 1 Gb/s line rate. Thus the switches were necessary in the topology to convert the 100 Mb/s link from the IXXAT 1588 boards to the 1 Gb/s link on the
Table 4.1: Baseline synchronization performance.

<table>
<thead>
<tr>
<th></th>
<th>Direct</th>
<th>Switches</th>
<th>Anue emulator and switches</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Synchronization Error (ns)</td>
<td>3</td>
<td>105</td>
<td>93</td>
</tr>
<tr>
<td>Standard Deviation of the Error (ns)</td>
<td>10</td>
<td>30</td>
<td>33</td>
</tr>
</tbody>
</table>

network emulator.

Figure 4-1: Topology 1 - 1588 traffic passing through the Anue System Network Emulator.

The second topology (Topology 2 shown in Figure 4-2) involved connecting each of the two identical IXXAT 1588 boards to a Netgear 1 Gb/s switch. Each switch had three links to a Spirent Smartbits 2000 1 Gb/s traffic generator. The two switches were also connected to one another. This topology was designed to inject traffic between two synchronized 1588 devices. Each Smartbits port was capable of transmitting data at line rate (1 Gb/s or 100% utilization), allowing us to congest the link between the two switches.
4.3 Objectives of the Phase 1 Research

Four experiments were performed to stress the 1588 clock synchronization between our two IXXAT 1588 boards. Two topologies were used throughout testing. Our experiments were designed with the following objectives:

- Observe the effect of latency probability distributions on 1588 device synchronization. The Anue System Network Emulator provides four different latency probability distributions: Fixed, Gaussian, Uniform, and Internet.

- Observe the impact of asymmetry in latency on the synchronization of 1588 devices.

- Observe the impact of increasing variance of Gaussian-distributed latency on the synchronization of 1588 devices.

- Observe the impact of network congestion and background traffic packet size has on 1588 clock synchronization.
4.4 Phase 1 Experiments

An experiment was designed for each of the four objectives. Each experiment stressed the 1588 clock synchronization between the two IXXAT 1588 boards. Three of the experiments involved varying latency algorithms and times to capture their effect on the clock synchronization. The last experiment applied congestion between the two IXXAT 1588 boards to stress the 1588 clock synchronization.

4.4.1 Experiment 1: Latency Probability Distribution

The purpose of Experiment 1 was to measure the impact of variable latency on 1588 device synchronization. Topology 4-1, shown in Figure 4-1, was used for this experiment. The Anue System Network Emulator provides four different latency probability distributions: Fixed, Gaussian, Uniform, and Internet. The “Internet” distribution is the term used by the Anue System Network Emulator to describe a Poisson distribution. For each of the probability distributions other than Fixed, the Anue requires the user to input a minimum and maximum delay time. For this experiment, we set the minimum delay to zero and varied the maximum delay. The experiment measured the effect each distribution had on the synchronization of the 1588 clocks.

All of the latency probability distributions offered by the network emulator introduce a variable amount of latency on the connection between the IXXAT 1588 boards. Figure 4-3 shows that all network latency distributions but Fixed had a dramatic effect on the 1588 clock synchronization. Focusing on the non-Fixed latency distributions, as the amount of latency was increased, the synchronization between the 1588 devices was significantly degraded. For the 1588 protocol to achieve high precision time synchronization it needs to determine how long a synchronization packet takes to be delivered. If that time is not static the 1588 devices constantly attempt to correct their clocks and, inevitably, become increasingly inaccurate.
Figure 4-3: The impact of the four latency probability distributions provided by the Anue System Network Emulator on the 1588 clock synchronization.
4.4.2 Experiment 2: Asymmetric Latency

The purpose of Experiment 2 was to measure the effect asymmetric latency had on the 1588 synchronized clocks. Topology 4-1, outlined in Figure 4-1, was used for this experiment. The Anue System Network Emulator was configured to use the Fixed probability distribution. The Anue allows the user to set the latency on both lanes of the Ethernet connection independently. This allowed us to apply asymmetric latency. The Anue only allows up to 1 ms delay to be applied to each lane. For each test in the experiment, each lane was increased by 2, 5, and 10 times the latency size of the other. To see the effect larger asymmetric delays might have on the clocks we performed the experiment with “master” latencies of 0.0001, 0.001, 0.01, and 0.1 milliseconds.

Figure 4-4 shows that as the latency grows, the clocks deviate from synchronization. IEEE 1588 has no way of detecting asymmetric latency. If the asymmetric latency is known, the correction can be made by a 1588 bridge or router [2]. In the experiment, an off-the-shelf 1 Gb/s switch was used and was not 1588-aware, nor did it have priority queuing enabled. This allowed us to exploit the 1588 protocols known weakness to see how much of an effect asymmetric latency had on 1588 clock synchronization.

4.4.3 Experiment 3: Variance of Gaussian Latency

The purpose of Experiment 3 was to measure the impact of varying the standard deviation of the Gaussian probability distribution on the 1588 clock synchronization. Topology 4-1 was used for this experiment. For this experiment we used the Gaussian distribution on the Anue System Network Emulator. The minimum delay was set to zero and the maximum delay was varied for this experiment. By increasing the maximum delay we could control the size of the standard deviation. The experiment varied the standard deviation from 0 to 1 ms.
Figure 4-4: The impact of asymmetric latency on the 1588 clock synchronization.
Figure 4-5 shows that a minor increase in variance of latency, has a significant impact on the clock synchronization. This experiment demonstrates that under a stressed network with varying latency, a 1588 device would not be able to maintain synchronization. Even with a 0.1 ms standard deviation, the 1588 clock synchronization was off by over 10 μs. For a protocol requiring sub-microsecond synchronization, this type of loss in precision is unacceptable. For 1588 to thrive, 1588 devices must be connected over a network that either implements 1588 or at least guarantees a high degree of consistency in the latency the packets experience.

Figure 4-5: The impact of increasing variance of Gaussian-distributed latency on the precision of the 1588 clock synchronization.
4.4.4 Experiment 4: The Effect of Congestion on Synchronization

The purpose of Experiment 4 was to measure the impact of heavy network congestion on the 1588 clock synchronization. Topology 4-2, outlined in Figure 4-2, was used for this experiment. To generate congestion-causing background traffic, three Spirent Smartbits 2000 traffic generator ports were connected to each switch. The Spirent Smartbits 2000 is capable of transmitting traffic at full 1Gb/s line rate. The Smartbits allows the user to select the speed at which to transmit data by varying the data field "%Utilization" (setting the percentage of the 1 Gb/s line rate used to transmit data).

In the experiment each Smartbits port was transmitting data to a different Smartbits port through the congested link. For each test run the individual Smartbits ports were set to transmit data at the same utilization rate. The experiment varied the congested link utilization rate from 10 to 300 percent.

In the 1 Gb/s Ethernet technology 64 and 1518 byte frames are the smallest and largest acceptable untagged frame sizes [11]. Untagged frames are frames that do not have a VLAN (Virtual Local Area Network) tag. The experiment was run with background congestion traffic with fixed frame lengths of 64, 128, 512, 1024, and 1518 bytes to find out if the different background traffic frame sizes had an impact on the clock synchronization. We chose the frames sizes to capture an even distribution from the minimum to the maximum length of a standard Ethernet frame.

Figure 4-6 shows that smaller frames have less of an impact on 1588 synchronization. Although high traffic of any type of frame could have a large effect on the synchronization, smaller frames tend to have a lesser impact. This is because if the switch's queue contains a 64-byte frame, and a 1588-byte frame enters the queue, the maximum time the 1588 frame will wait is the time it takes for the current 64-byte frame to transmit. If a 1518-byte frame is in the queue instead of the 64-byte frame, the 1588 frame could have to wait almost 24 times longer. Increasing the frame size
results in a larger standard deviation of the variable latency. Experiment 3 supports this claim by showing the larger the standard deviation of the variable latency, the less precise the 1588 clock synchronization becomes.

Figure 4-6: The impact of network congestion caused by background traffic on 1588 clock synchronization.
Chapter 5

Phase 2

Our analysis of previous work by the 1588 community and the results of Phase 1 motivated further study of the performance of TC bridges and Enterprise bridges on the same fabric when asymmetric latency and congestion are present. We studied a variety of TC bridges and Enterprise switches both together and on separate fabrics to find the impact TC bridges have on performance of a real 1588 networks.

The topologies used in this phase were based on the ones employed in Phase 1 and include additional 1588 devices, together with TC and Enterprise Ethernet Bridges. The overall goal is to analyze how TC bridges work in a mixed fabric of Enterprise Ethernet bridges, specifically we:

- Analyze how congestion from other sources on the network effect the 1588 fabric with TC bridges only and a mixture of TC bridges and Enterprise Ethernet bridges (Experiment 1).

- Analyze how latency (both symmetric and asymmetric) effect the 1588 fabric with TC bridges only and a mixture of TC bridges and Enterprise Ethernet bridges (Experiment 2).

At a broader level, the goal of the project was to determine whether Ether-
net VLANs, priorities, and other existing queuing solutions of Enterprise Ethernet switches can be used to enhance time synchronization of 1588 devices. If existing Enterprise Ethernet-based solutions maintain adequate performance, then some of the cost associated with replacing an entire network with 1588 aware devices could be avoided.

5.1 Experiment 1: The Effect of Congestion on Synchronization with Transparent Clocks and Enterprise Ethernet Switches

The purpose of Experiment 1 is find out whether or not Transparent Clocks are necessary for maintaining high clock synchronization accuracy in a 1588 environment. Since many adopters of 1588 technology already have existing enterprise Ethernet switches, it is important to determine if these switches can properly be used in a 1588 environment. Our experiment has two topologies to test the need for TCs. The first part tests end-to-end unicast congestion and the second part tests broadcast congestion.

Figure 5-1 and 5-2 outlines the generic setups for Experiment 1. Figure 5-1 has a traffic generator connected to Switch B while Figure 5-2 has two traffic generators connected to Switches A and C. For the testing we used the two IXXAT BF518 EZ-Kit 1588 development boards from the preliminary research and two Meinberg Lantime M600 1588 endpoints. Each 1588 device has a Pulse Per Second (PPS) clock output connected to a Lecroy SDA 950 scope for measuring clock accuracies. Switches A, B and C are placeholders for different switches used for testing. Table 5.1 defines explains the topologies used in the end-to-end unicast traffic experiment. The term Dominant in the graphs are used to explain that there is a 2:1 ratio of the particular switch type in the topology. The term Core in the graphs is used to denote that the
Figure 5-1: Generic Topology for Experiment 1 - 1588 traffic passing through the three switches A, B and C. The Meinberg Lantime M600 connected to Switch A will be the master in all of testing for experiments. A Spirent Smartbits 2000 was used to generate broadcast traffic to congest all devices on the networks.

Figure 5-2: Generic Topology for Experiment 1 - 1588 traffic passing through the three switches A, B and C. The Meinberg Lantime M600 (labeled M) connected to Switch A will be the master in all of testing for experiments. A Spirent Smartbits 2000 was used to generate unicast traffic to congest all switches on the networks.
particular switch type is in the center of the topology.

<table>
<thead>
<tr>
<th>Table 5.1: Devices used in Experiment 1.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>Eth-Core Eth-Dom</td>
</tr>
<tr>
<td>Figure 5-3</td>
</tr>
<tr>
<td>TC-Core Eth-Dom</td>
</tr>
<tr>
<td>Figure 5-4</td>
</tr>
<tr>
<td>Eth-Core TC-Dom</td>
</tr>
<tr>
<td>Figure 5-5</td>
</tr>
<tr>
<td>TC-Core TC-Dom</td>
</tr>
<tr>
<td>Figure 5-6</td>
</tr>
</tbody>
</table>

The switches we used for the TCs were Vitesse and Hirschmann Transparent Clocks. For the enterprise Ethernet switches we used two Fulcrum Monaco cut-through switches with priorities enabled that provides near wire speed transmission on the network. The term “cut-through” refers to the way the switch transmits and receives frames. Once a cut-through switch receives the header of a packet and knows its destination, it will start forwarding the frame before the entire frame has been received. The Ethernet switches used in this topology are switches that would be seen in high-end enterprise switching environments such as data clusters.

5.1.1 Experiment 1: Congested Switch Links via End-to-End Unicast Traffic

Figure 5-2 generalizes the end-to-end unicast traffic topology. These topologies are shown in Figures 5-3, 5-4, 5-5 and 5-6. In the figures, a 1588 Grand Master Clock
and a slave are directly connected to switch A. The slave connected to switch A is labeled in the topology as 2. There is a 1588 device labeled as 3 directly connected to switch B. Switch B is connected to switches A and C, creating a 3 hop topology between the Grand Master Clock labeled as M and the 1588 device labeled as 4. For graphing purposes, in all of Experiment 1 topologies, these devices will remain in the same positions and be denoted by the same labels.

![Diagram](image)

Figure 5-3: *Eth-Core Eth-Dom:* Enterprise Ethernet Core and Enterprise Ethernet Dominant Topology for Experiment 1.

![Diagram](image)

Figure 5-4: *TC-Core Eth-Dom:* Transparent Clock Core and Enterprise Ethernet Dominant Topology for Experiment 1.
Figure 5-5: *Eth-Core TC-Dom*: Enterprise Ethernet Core and Transparent Clock Dominant Topology for Experiment 1.

Figure 5-6: *TC-Core TC-Dom*: Transparent Clock Core and Transparent Clock Dominant Topology for Experiment 1.
In this experiment unicast traffic was transmitted across the three hop topology between the two Smartbits 2000s. We used the Lecroy SDA 950 to compare the Pulse Per Second (PPS) output of all four 1588 devices. The Lecroy was set to gather the pulses and compare them to the Grand Master Clock PPS. It would then record the standard deviation and mean synchronization error for each 1588 device. We measured 300 pulses per data point, which is roughly 5 minutes of capturing time. We chose to take only 300 data points because larger sample sizes were returning the same results. In all future experiments we gathered data in exactly the same way as in this one, unless stated otherwise.

The results of the topologies shown in Figures 5-3, 5-4, 5-5 and 5-6, are shown in Figures 5-7, 5-8 and 5-9.

The results shown in Figures 5-7, 5-8 and 5-9 are for the slave at positions 2, 3 and 4 respectively in the topology figures 5-3 through 5-6. In all figures the mean synchronization error in microseconds is on the y-axis and the utilization of the 1Gb/s links between the switches is on the x-axis. Confidence intervals are drawn at each data point. In Figure 5-7, as the switches become overwhelmed with traffic, all topologies maintain close to sub-microsecond accuracy. The traffic should have no effect on the link between the slave in position 2 and the Grand Master Clock because they are both connected to the same switch. The graph shows that some variance in accuracy is introduced when the switches become overwhelmed. This can be attributed to the switch buffers being overwhelmed.

Figure 5-8 shows a dramatic effect on the mean synchronization error for two topologies: TC-Core TC-Dom (Figure 5-6) and Eth-Core TC-Dom (Figure 5-5). At 50% utilization, the TC-Core TC-Dom (Figure 5-6) topology loses all of its synchronization accuracy having over a 600 microsecond mean synchronization error with almost an 800 microsecond standard deviation. This can be attributed to the fact
Figure 5-7: Experiment 1- mean synchronization error for 1588 device 2, all nodes on the network were congested with broadcast Ethernet frames.
that in Figure 5-6 the master (M) to slave (3) link has to travel through two Transparent Clocks A and B and no cut-through Ethernet switches. The TC buffers were being overwhelmed at 50% utilization and packets were being dropped. The other topologies were not effected at 50% utilization because the Ethernet cut-through switch prioritizes the 1588 frames and moves them to the front of the exit queue in the switches buffer. Since all of the other topologies have at least one cut-through switch between the master (M) and the slave device located in position 3, 1588 packets were not dropped. At 100% utilization only topologies 5-3 and 5-4, where the master was directly connected to a cut-through switch, maintained clock precision. Both TC-Core TC-Dom (Figure 5-6) and Eth-Core TC-Dom (Figure 5-5) had a TC directly connected to the master. The TC buffers would become overwhelmed and traffic would be dropped causing loss of synchronization. This is consistent with the theoretical prediction where the master to slave link is an order of ten times more important than the slave to master link [27].

Figure 5-9 (slave in position 4) also shows a dramatic effect on the mean synchronization error for three topologies: TC-Core TC-Dom (Figure 5-6), Eth-Core Eth-Dom (Figure 5-3) and Eth-Core TC-Dom. Like the device in position 3, the device in position 4 experiences loss of synchronization in the TC-Core TC-Dom (Figure 5-6) topology when 50% utilization of the links between the switches is achieved. Position 4 differs from 3 in that synchronization is lost for all topologies when 100% utilization is achieved except TC-Core Eth-Dom (Figure 5-4). This is because the TC-Core Eth-Dom (Figure 5-4) has a cut-through switch connected to each of the Smartbits 2000s at positions A and C. When switches get overwhelmed they will drop packets at the end of their buffer queues. Since the 1588 frames move to the front of the queue they are sent out immediately and never get dropped. This causes all 1588 traffic to go end to end without losing synchronization. All other topologies have a
Figure 5-8: Experiment 1- mean synchronization error for 1588 device 3, all nodes on the network were congested with broadcast Ethernet frames.
Smartbits 2000 connected to at least one TC, causing that TC to be the bottleneck.

Figure 5-9: Experiment 1- mean synchronization error for 1588 device 4, all nodes on the network were congested with broadcast Ethernet frames.

5.1.2 Experiment 1: Congested Links via Broadcast Traffic

Figure 5-1 generalizes the broadcast traffic topologies. These topologies are clearly defined in Figures 5-10, 5-11, 5-12 and 5-13. They are almost exactly the same topologies defined in the first topology generalized in Figure 5-2, except for the position and traffic generated by the Smartbits 2000. In this topology there is a Smartbits 2000 connected to switch B, in the core of the topology instead of at both ends of the topology. The Smartbits 2000 traffic transmits broadcast traffic to all of the devices.
on the network, and for this experiment the percent utilization of the traffic is varied.

Figure 5-10: *Eth-Core Eth-Dom*: Enterprise Ethernet Core and Enterprise Ethernet Dominant Topology for Experiment 1.
Figure 5-11: **TC-Core Eth-Dom**: Transparent Clock Core and Enterprise Ethernet Dominant Topology for Experiment 1.

Figure 5-12: **Eth-Core TC-Dom**: Enterprise Ethernet Core and Transparent Clock Dominant Topology for Experiment 1.
Figure 5-13: *TC-Core TC-Dom*: Transparent Clock Core and Transparent Clock Dominant Topology for Experiment 1.
The results shown in Figures 5-14, 5-15 and 5-16 are for the slaves at positions 2, 3 and 4 respectively in the topology Figures 5-10 through 5-13. In all figures the mean synchronization error in microseconds is on the y-axis and the percent utilization of the 100Mb/s links is on the x-axis. Confidence intervals are drawn at each data point. In Figures 5-14, 5-15 and 5-16 we see that all topologies have nearly the exact same results with very large standard deviations. The 1588 devices were unable to handle broadcast traffic larger than 20% utilization of a 100 Mb/s link. Due to these hardware limitations we were only able to test up to 20% utilization of a 100 Mb/s link. These results show that under moderate congestion a Transparent clock provides little to no benefit.

Figure 5-14: Experiment 1 synchronization of 1588 device at location 2 on all topologies in a network congested with broadcast Ethernet frames.
Figure 5-15: Experiment 1 synchronization of 1588 device at location 3 on all topologies in a network congested with broadcast Ethernet frames.
Figure 5-16: Experiment 1 synchronization of 1588 device at location 4 on all topologies in a network congested with broadcast Ethernet frames.
5.2 Experiment 2: The Effect of Asymmetric Latency on Transparent Clocks and Enterprise Ethernet Switches

The purpose of Experiment 2 was to measure the effect asymmetric latency had on the 1588 synchronized clocks when in TC only, enterprise Ethernet switches only and mixed TC and enterprise Ethernet switching environments. Figure 5-17 is the generic topology used for these experiments. The Anue System Network Emulator was configured to use the Fixed probability distribution. The Anue allows the user to set the latency on both lanes of the Ethernet connection independently. This allowed us to apply asymmetric latency. The Anue only allows up to 1 ms delay to be applied to each lane. As in our preliminary research for each test in the experiment, each lane was increased by 2, 5, and 10 times the latency size of the other. To see the effect larger asymmetric delays might have on the clocks we performed the experiment with “master” latencies of 0.1 milliseconds.

Figure 5-17 outlines the generic setup for Experiment 2. For the testing we used the two IXXAT BF518 EZ-Kit 1588 development boards from the preliminary research, and two Meinberg Lantime M600 1588 endpoints. Switch A to B are placeholders for different switches used for testing. The topologies we looked to experiment with are show in Table 5.2.

The switches we used for the TCs were Vitesse and Hirschmann Transparent Clocks. For the enterprise Ethernet switches we used an HP ProCurve 2510-48 and a Broadcom BCM switch. The Ethernet switches used in this topology are switches that would be seen in enterprise switching environments.

Figure 5-18 shows the topology used for the all enterprise Ethernet topology. The Broadcom (B) and HP (A) switches are indicative of what would be seen in enterprise situations. Figure 5-19 shows that as the asymmetry between the two connections grows the mean synchronization error of the switches becomes greater. At five times
Figure 5-17: Generic Topology for Experiment 2 - 1588 traffic passing through the Anue System Network Emulator. The Meinberg Lantime M600 connected to switch A will be the master in all of testing for experiments.

Figure 5-18: Enterprise Ethernet only Topology for Experiment 2 - 1588 traffic passing through the two enterprise Ethernet switches and the Anue System Network Emulator.
the latency difference, the accuracy of the 1588 devices are not equivalent across the 1588 network, showing an inconsistency in the buffers of the Ethernet switches. 1588 device slave 2 (Meinberg) also shows asymmetric delays from master to slave and slave to master. 1588 device slave 4 (IXXAT) maintains symmetric accuracy over the network. Overall the mean synchronization error is inconsistent and erratic between device 2 and 4 even though they are connected to the same switch having up to a 50 microsecond difference with one another.

Table 5.2: Devices used in the asymmetric latency experiments.

<table>
<thead>
<tr>
<th></th>
<th>Switch A</th>
<th>Switch B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eth-Only</td>
<td>Eth</td>
<td>Eth</td>
</tr>
<tr>
<td>Figure 5-18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Eth-TC</td>
<td>Eth</td>
<td>TC</td>
</tr>
<tr>
<td>Figure 5-20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TC-Eth</td>
<td>TC</td>
<td>Eth</td>
</tr>
<tr>
<td>Figure 5-22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TC-Only</td>
<td>TC</td>
<td>TC</td>
</tr>
<tr>
<td>Figure 5-24</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 5-20 shows the topology used for the mixed topology of enterprise Ethernet and TC. The Broadcom (A) is connected to the master (M) and slave 3, while the Hirschmann TC (B) is connected to slave 2 and 4. As with Figure 5-19, Figure 5-21 shows that as the asymmetry between the two connections grows the mean synchronization error of the switches becomes greater. At all points in the graph, the mean synchronization error is symmetric on both master and slave sides. These results are different from our Ethernet only topology (5-19) because Figure 5-21 shows more consistency on our master to slave delays. It is also observed that at five times the
Figure 5-19: Enterprise Ethernet only results for Experiment 2 - The impact of asymmetric latency on the 1588 clock synchronization in an enterprise Ethernet topology.

Figure 5-20: Enterprise Ethernet to TC Topology for Experiment 2 - 1588 traffic passing through an enterprise Ethernet switch, TC and the Anue System Network Emulator.
latency difference the mean synchronization error in Figure 5-21 is only 200 microseconds while in the Ethernet only topology of Figure 5-18 we find results 5-19 as high as 250 microseconds. This can be attributed to inconsistent wait times in the buffers of the Ethernet switches. These results are consistent with the results from a paper by R.L. Scheiterer et al., where the author concludes the master to slave link is an order of ten times more important than the slave to master link [27]. Adding a TC (B) on the side of the slaves 2 and 4 in Figure 5-20 corrects the time stamps of the frames that were transmitted between the master and the slaves after the asymmetric delay injected by the Anue, making our results more consistent.

Figure 5-21: Enterprise Ethernet to TC results for Experiment 2 - The impact of asymmetric latency on the 1588 clock synchronization in a mixed topology.
Figure 5-22: TC to Enterprise Ethernet Topology for Experiment 2 - 1588 traffic passing through the a TC, enterprise Ethernet switch and the Anue System Network Emulator.

Figure 5-22 shows the topology used for the mixed topology of enterprise Ethernet and TC. The Vitesse TC (A) is connected to the master (M) and slave 3, while the Broadcom (B) is connected to slave 2 and 4. As with Figures 5-19 and 5-21, Figure 5-23 shows that as the asymmetry between the two connections grows the mean synchronization error of the switches becomes greater. At five times the latency difference, the accuracy of the 1588 devices is not equivalent across the 1588 network, showing the same inconsistency in the buffers of the Ethernet switches as Figure 5-19. Overall the mean synchronization of the topology in Figure 5-22 is similar to the all Ethernet topology in Figure 5-18 because the superior master (M) to slave 2 and 4 links have an Ethernet switch B between the asymmetric latency injected by the Anue and the slaves.

Figure 5-24 shows the topology used for the TC only topology. The Vitesse TC A is connected to the master (M) and slave 3, while the Hirschmann TC B is connected to slave 2 and 4. Figure 5-25 shows that as the asymmetry between the two connections grows the mean synchronization error of the switches becomes greater. The results are almost identical to Figure 5-21 because the master to slave link has the TC after the injected asymmetric latency.
Figure 5-23: TC to Enterprise Ethernet results for Experiment 2 - The impact of asymmetric latency on the 1588 clock synchronization in a mixed topology.

Figure 5-24: TC only Topology for Experiment 2 - 1588 traffic passing through the two TCs and the Anue System Network Emulator.
Figure 5-25: TC only results for Experiment 2 - The impact of asymmetric latency on the 1588 clock synchronization in a TC only topology.
The results show that the master to slave link is far more important than the slave to master in maintaining synchronous symmetric mean synchronization error. Although having Transparent Clocks makes the network more reliable when properly placed, they do not solve the problems an asymmetric connection can have on a 1588 network.
Chapter 6

Conclusion and Future Work

Phase 1 experiments have shown that under high network latency variance 1588 devices struggle to maintain clock synchronization. When asymmetric latency is experienced by 1588 devices, it becomes significantly difficult to uphold precise clock synchronization. Our tests also demonstrate that variable latency makes it very challenging for a 1588 device to preserve clock synchronization. In summary, 1588 devices perform exceptionally well under fixed latency circumstances but struggle under variable and asymmetric latency situations.

Phase 2 research builds on Phase 1 and shows Transparent Clocks on small topologies help to maintain sub-microsecond clock synchronization in the face of congestion. We also found under congestion, Transparent Clocks without prioritization, cannot maintain high accuracy clock synchronization. Having cut-through Enterprise Ethernet switches connected to high congestion endpoints with priorities enabled results in better performance in congested networks, however, they do not maintain the same sub-microsecond accuracy TCs have without congestion present. The results also shows that when 1588 devices themselves are congested directly via broadcast frames, the TCs had little to no effect on maintaining clock synchronization.

Asymmetric latency results in Phase 2 Experiment 2 show that Transparent Clocks
help to reduce variance of the 1588 devices.

Although Transparent Clocks provide many benefits for maintaining high precision clock synchronization, alone in a congested environment they are inadequate without prioritization. Our results show that strategically placing cut-through Enterprise Ethernet switches with priorities enabled in highly congested locations resolves many issues that congestion presents. Additionally, cut-through switches with priorities enabled also provide adequate accuracy when compared to TCs. Ultimately, an all cut-through Enterprise Ethernet network is adequate for maintaining sub-microsecond accuracy time synchronization with or without congestion in a 1588 environment.

For future work we plan to test these experiments using different 1588 boards capable of handling utilization larger than 20% of the connection. We also plan to test these experiments with prioritization enabled Transparent Clocks. We also plan to program a Transparent Clock software stack on the Fulcrum Enterprise Ethernet switch because we believe the combination of TC and Enterprise Ethernet features is the most optimal approach.
Bibliography


[11] IEEE Standard 802.3-2002 subclauses 4.2.3.3, 4.2.4.2.2, 4.2.9, 3.2.8, 4.2.3.1, 4.2.4.1.2, 4.2.9, and 46.3.3.1.


