Final Year Report - Investigation into the Precision Time Protocol

James Cox
Department of Electrical and Electronic Engineering

University of Bath

March 19, 2014

Abstract

Abstract goes here

Contents

Acronyms

CSAC

CSMA/CD Carrier Sense Multiple Access with Collision Detection

GM Grandmaster

GPS Global Positioning System

IEEE Institute of Electrical and Electronic Engineers

ITU International Telecommunication Union

LAN Local Area Network

NERC North American Electric Reliability Company

NTP Network Time Protocol

PPS Pulse per Second

PTP Precision Time Protocol

PTPd PTP Daemon

SNTP Simple Network Time Protocol

TDEV Time Deviation

UTC Coordinated Universal Time

List of Figures

List of Tables

1 Introduction

In several applications, maintaining a high level of time accuracy is very important. This may either be a physical timestamp or that there is a need to synchronise process occuring at specific times. Some examples of these applications are: telecommunications, the automation industry, or the power transmission industry.

There are currently only a few methods to realise this, which depends heavily on the application. For example, if there is space and the budget, a high accuracy atomic clock (CSAC) could be used. If this is not possible, then a Global Positioning System (GPS) receiver could be used and time can then be synchronised to the accurate clocks onboard the GPS satellites.

This leads to some issues when heavily relying on GPS for time synchronisation. There would be catastrophic consequences in some applications if timing is not kept accurate. The issues relate to the ease of jamming of a **GPS receiver!** (**GPS receiver!**). [reference]

Due to the jammability of GPS, there must be either be a backup solution or an alternative method of time synchronisation in order for accurate timing to still be possible.

One way of realising this is by using a distributed timing system, such as Network Time Protocol (NTP) or Precision Time Protocol (PTP). This would allow for nodes to be able to synchronise their clocks with a much more accurate time source without having to rely on GPS

1.1 How PTP Works

Explain PTP Briefly

1.2 Project Description

This project aims to investigate PTP performance on a heavily used Ethernet network, and to attempt to quantify PTP performance using packet metrics.
There are also some deliverables as part of the Final Year Project. These include: an interim report, a log book, a final year report and a poster. These deliverables, along with the sub tasks involved in order to complete them have been detailed in the Gantt chart, as seen in Appendix A.

The following project objectives have been identified:

Learn about PTP and other work in relation to the protocol
This stage would occur at the beginning of the project to understand how PTP works. This is important so work can then be carried out to investigate PTP performance on a network.

Collect PTP Data
In parallel with the above, PTP data can be collected. This will be monitoring the performance of PTP across the network as well as how using multiple types of grandmaster/slaves affect the performance. Different clock locations in the network will also be considered.

Implement some packet metric scripts
To be able to understand the performance of the network, some packet metric scripts will be created. A suitable language will be chosen once this part of the project begins.

Determine packet performance using these scripts
Multiple window sizes and types of metric will be used to quantify network performance.

Test Chronos' equipment and provide feedback
As Chronos has provided this project with some equipment, this equipment will also be thoroughly tested and any information gathered can be passed to them once the project is completed.

2 Literature Review

With the following objectives and tasks in mind, a literature review was performed with some suitable documentation: mainly packet metric related, but also on cryptographically signing PTP packets.

The reports below will be discussed in some detail:

Definitions and terminology for synchronization in packet networks [?] A standard regarding different packet metrics that could be used in order to try and quantify network delay.

Prevention of Packet Collisions [?] A journal article describing an algorithm that aims to prevent packet collisions in an Ethernet network.

2.1 Lit Review for Packet Metric

Lit review for packet metric

2.2 Other Relevant Reading

There was other relevant reading performed in the first week of the project to do with cryptography and how packet collisions can be prevented. - cryptography ?

3 Project Methods

Based on the objectives mentioned previously, the project can be split into three distinct sections:

Data Collection
This part of the project will involve collecting PTP timing data on the university network. It will consist of using a number of different clock types and locations on the network.

Packet Methods
This section will mainly involve the implementations of the packet metric scripts based on the referenced report above. Focus on the implementation will be made in this section rather than the metrics themselves.

Calculating/Analysing Results
Once the metrics have been implemented fully, there needs to be some supplementary scripts written to process some of this data.

4 Data Collection

The first step to perform with this part of the project is to work out what hardware is available. The following hardware was identifed as being available to use for the duration of this project.

- Hardware Grandmaster - Chronos TimePort [?]
- Hardware Slave - Chronos Syncwatch [?]
- Hardware Slave - Beaglebone Black []
- Software Grandmaster - PTP Daemon (PTPd)
- Software Slave - PTPd

4.1 Hardware - Timeport [?]

4.1.1 Description

The Chronos CTL4540 Timeport is a low powered portable device that is able to maintain its time to a high accuracy when disconnected from a synchronisation source. It is able to maintain accuracy within a couple hundred nanoseconds without needing to be connected to GPS. It also has an internal LiPo battery. This enables the device to be used to transport and measure time.

With the above features in mind, it is thus suited for a number of markets, including the power industry and telecommunication network operators. It can also be used to correct for any time errors caused by any cabling

(a) Chronos TimePort Outside

Figure 1: Chronos TimePort Labelled Diagrams

or equipment.

Typical methods of doing this would involve using a Caesium atomic clock [REF] or setting up a GPS attenna and connecting this to some other equipment. The TimePort is best suited over these two operaitons because it is much lower power and much more transportable than an atomic clock. It also removes the requirement of GPS equipment.

Appendix **??** shows the full specifications of the CTL4540 TimePort. Below are a few labelled photos of the clock.
The difference between the release TimePort and the TimePort that will be used in this project is that the firmware on the TimePort is bleeding edge. With that in mind time needs to be allocated to allow for any issues that the clock may have. The university has close links with Chronos thus it should be straightforward to either get our issues solved or to receive a new TimePort.

This clock will mainly be kept in the same position on the network and will acct as a Grandmaster.

In terms of documentation there is not much available for this device apart from some emails sent between Chronos and Dr Robert Watson. Therefore Appendix **??** shows some documentation put together for my own use during this project. The documentation includes details on how to interface with the clock and a list of basic commands.

To access the device it needs to be accessed locally over a USB to Serial connection. SSH is unavailable as the control port has not been implemented yet.

4.1.2 How to Set Up the TimePort

- notes in logbook. - will complete this section Thursday. - screenshots too

4.2 Hardware - Chronos Syncwatch [?]

4.2.1 Description

The Chronos Syncwatch is a hardware slave clock used to synchronise time in a number of different applications. It operates in all of the current synchronisation technologies. SyncE, ESMC, PTPv2, 1PPS+TOD, 1PPS, Frequencies(64k-200MHz), T1 & E1 protocols and interfaces are supported.

It can be used on both legacy and modern Ethernet/IP networks. It can simultaneously operate on a number of the protocols above. It can also operate in both local and remote modes.

It is a small modular device with a simple user interface. It also integrates with with Symmetricom's TimeMonitor software.

The device markets include telecommunications, TV and radio broadcasting, and the power industry.

The table shown in Appendix **??** details the Syncwatch specifications. The figures below show labelled diagrams of the inside and outside of the Chronos Syncwatch.

(a) Chronos Syncwatch Outside

(b) Chronos Syncwatch Inside

Figure 2: Chronos Syncwatch Labelled Diagrams

Figure 3: Labelled BeagleBone Black

This product is similar to the TimePort in the fact that there isn't much documentation around for it. Therefore Appendix **??** shows the documentation written up for the Syncwatch.

The Syncwatch will be mainly kept in the upstairs Level 3 Communications lab as it is a larger device. As this device is a release product, all of the ports are working. Therefore the syncwatch can be set up via SSH or using the program. This is all explained in the documentation in Appendix **??**.

4.2.2 How to Set Up the SyncWatch

- notes in logbook. - will complete this section Thursday. - screenshots too

4.2.3 Hardware - Beaglebone Black [?]

4.2.4 Description

The Beaglebone Black is a hardware device but it is running a software PTP Daemon (called PTPd).

In terms of hardware capabilities it has an ARM Cortex A-8 processor with 512MB of DDR3 RAM. It runs a cut down version of Linux called Angstrom Linux. It has Ethernet connectivity and runs off of a 5V DC supply.

As it runs Linux and can be connected to the network, an SSH server has been set up on it with a static IP address. This made it easy to start the PTP daemon.

Below (Figure 3 is a labelled picture of the BeagleBone Black.
The Beaglebone will be a useful device to use as a slave clock because of its portability. It would be able to be placed anywhere on the network without any disruption to that particular lecture room or lab space.

4.2.5 How to Set up the Beaglebone Black

When the project was started, Robert Watson had the BeagleBone Black working with PTPd already, so there was only some work to be done in order to automate the process.

To set up the Beaglebone Black:

1. Plug in the Beaglebone Black to the 5V adapter.
2. Plug in the Ethernet cable
3. Once the Beaglebone boots you can then access the device over SSH.

Type:

```
ssh jac50@eepc-rjw-beaglebone.bath.ac.uk
```

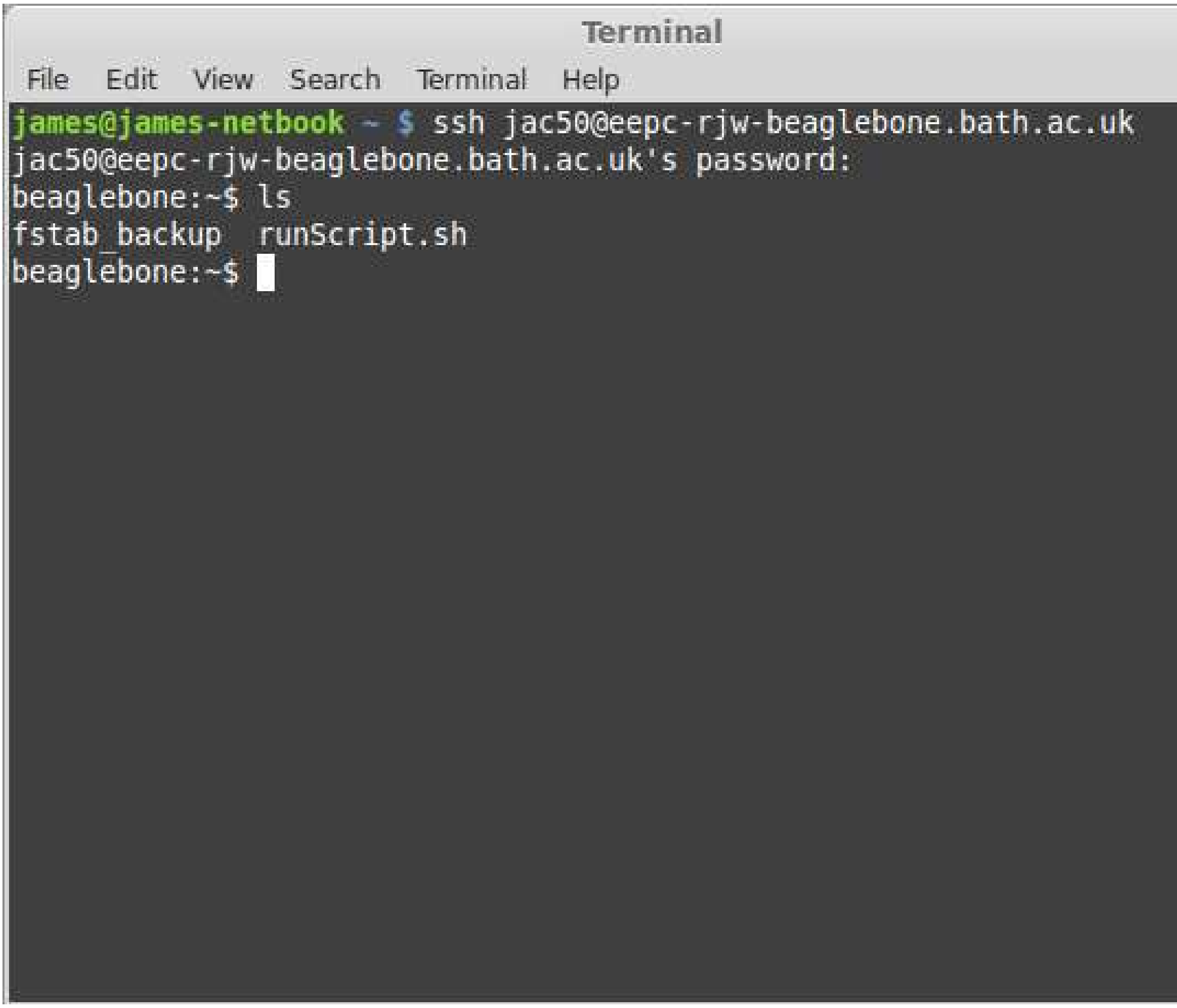

Figure 4: SSH to Beaglebone

to log in, replacing jac50 with the username on the device. The users on the device were eerjw, jac50, and root.

The screenshot below (Figure 4) demonstrates this. The ls command was typed to show that the connection was successful.
Once SSH'd into the device, then the device can be accessed like any other linux machine. Note however that there is a restricted command set.

When the BeagleBone boots, it was required that the SD card used to store the test data on would be automounted and that the PTP daemon automatically runs.

Several attempts in trying to automount the SD card using conventional means such as adding in an entry to fstab were attempted, but this did not work.

The method in getting round this was by creating a script in /etc/init.d. Any script located in that folder will automatically be loaded once the device boots. The PTP daemon was also run from this same script. The script would also need to automatically name the data file or the data would be overwritten every time the device was turned on. A convention of *timeport_YYYY_MM_DD* was decided.

The full bash script is shown below, in Code Extract **??**

```
#!/bin/bash
mount /dev/mmcblk1p1 /mnt/sd

\_date=$(date +"%Y_%m_%d")
sudo /home/eesrjw/ptpd2 -i eth0 -C -S -g -d 17 -V > /mnt/sd/eesrjw/timeport_$_date.txt
```

Listing 1: Bash Script in init.d for Beaglebone Black

4.3 Software - PTPd

Same as above

4.4 Data Collection Overview

The summary table below shows what hardware is available, based on class, movability, and ease of transferring data off of it.

Table 1: Hardware Summary

Clock ID	Name	Type	Role	Location Mobility	PTP IP	Control IP
001	Chronos TimePort	Hardware	Grandmaster	Mobile	TimePort	USB over Serial
002	Chronos SyncWatch	Hardware	Slave	Mobile	syncwatchptp	eepc-rjw-syncwatc
003	BeagleBone Black	Hardware	Slave	Mobile	beaglebone	eepc-rjw-beaglebo
004	PTPd_Desktop	Software	Slave	Fixed	...	...
005	PTPd_Netbook	Software	Slave	Mobile	...	...

The clock ID was used with internal documentation to know which clock was used where. Note that the IPs listed are just part of the full name. To access one of them on the university network, add the prefix eepc-rjw- and the suffix .bath.ac.uk.

4.5 Locations for Clocks

4.6 Test Sheets

explain brief reason as to why test sheets will be used. stored in excel
post example of one of the overall test sheets and the individual test sheets.

4.7 Testing Schedule

This part of the project will run in parallel with the implementation stage of the packet metrics, as this does not rely on them being completed. *list types of tests wanting to be run *tabulate locations + clock types? (rough dates/times maybe)

5 Packet Methods

* explain briefly the packets are used, and refer to report *explain which packets will be chosen (why they were chosen) *list all metrics that will be implemented

5.1 Choosing a suitable language

*ranking tables to choose the correct language

5.2 General Packet Implementation

5.2.1 TDEV and TDEV derivatives

5.2.2 MATIE and MAFE derivatives

5.2.3 Overall Packet Script

5.2.4 Testing

5.2.5 Optimisation

5.3 Calculating Results

*explain workflow of collecting data

5.3.1 Other Utility Scripts

explain what utility scripts otherwise not mentioned above that will be used.
Rsync Script
Awk Data Parser

6 Results

*explain results

7 Milestones

8 Discussion

9 Conclusion

10 Acknowledgements

Appendix A Gantt Chart and Table

Figure 5: Gantt Chart

Appendix B Chronos CTL4540 TimePort Specification

IPPS	200 nanoseconds over 8 hours (±10 °C temp change)
Holdover	100 nanoseconds ove 4 hours (±10 °C temp change)

Inputs

+5V DC:	MiniB USB
GPS antenna:	SMA
Ethernet (PTP and SNTP/NTP):	RJ45 10/100
Ethernet (management):	RJ45 10/100
1PPS (phase 2):	BNC

Outputs

1PPS:	BNC
Frequency 1: 2.048 MHz, 10 MHz	BNC G.703
Frequency 2: 2.048 MHz, 10 MHz	BNC G.703
IRIG-B:	BNC
RS232:	9 way D-Type 9600 band
RS442:	15 way D-Type 9600 band
Ethernet (PTP and SNTP/NTP) (Max 10 clients):	RJ45 10/100
Ethernet (management):	RJ45 10/100

Environmental

Operating Temperature:	0 °C to +50 °C
Maintain holderover tolderance down to:	−10 °C for 15 minutes
Storage temperature:	−20 °C to +80 °C

Physical

Size:	190 x 57 x 170mm (WxHxL)
Weight:	1150g

Appendix C TimePort Documentation

Appendix D Introduction

Appendix E Scope

Appendix F Diagram of a TimePort

Appendix G How to set the TimePort up as a Grandmaster

Appendix H Problems and Solutions

Appendix I Conclusion

Appendix J Test Sheet Example

	Test: Timeport_to_Software Test One			
Test Name:	TimePort_To_Software Test One			
Test ID:	001			
Test Date	2014-02-27			
File Name:	RawData.txt			
Directory:	./PTPData/TimePort_To_Software_Test1			
Start Time:	1037			
End Time:	2200			
Clock #1 Type:	Hardware		**Clock #2 Type:**	Software
Clock #1 Name:	TimePort_1		**Clock #2 Name:**	PTPd_Netbook
Clock #1 Model:	TimePort		**Clock #2 Model:**	PTPd
Clock #1 Location:	Watson's Office		**Clock #2 Location:**	2E 2.13
Network Activity:	Normal			
Test Description:	An initial test to collect data to supplement the example data already received.			
Comments	1342: Data seems to be collecting fine. 3hrs20mins: 45MB			

Appendix K Test Sheet Summary Sheet

Test Number	Directory	Master	Slave	Location Master	Location Slave	Sta
001 Finished	27/02/14	TimePort-To-Software-Test1	TimePort_1	PTPd_Netbook	2E ..	2E
002 Finished	28/02/14	TimePort-To-Software-Test2	TimePort_1	PTPd_Netbook	2E ..	2E
003 In Progress	28/02/14	TimePort-To-Software-Test3	TimePort_1	PTPd_Desktop	2E ..	2E
004 Finished	03/03/14	TimePort-To-Software-Test4	TimePort_1	PTPd_Netbook	2E ..	2E
005 Finished	03/03/14	TimePort-To-Software-Test5	TimePort_1	PTPd_Netbook	2E ..	Lib
006 Finished	03/03/14	TimePort-To-Beaglebone-Test1	TimePort_1	Beaglebone_1	2E ..	2E
007 Finished	04/03/14	TimePort-To-Software-Test6	TimePort_1	PTPd_Netbook	2E ..	2E
008 In Progress	05/03/14	TimePort-To-Beaglebone-Test2	TimePort_1	Beaglebone_1	2E ..	2E
009 In Progress	05/03/14	TimePort-To-Software-Test7	TimePort_1	PTPd_Netbook	2E ..	2E

Appendix L Main Packet Metric Script

Appendix M TDEVAllMethods

Appendix N MATIEAllMethods