HELSINKI UNIVERSITY OF TECHNOLOGY
Networking Laboratory
Telecommunications Engineering

EVALUATION OF TIME SYNCHRONISATION ON A RESEARCH NETWORK

Master's Thesis

Carlos Muñoz Gahete

Networking Laboratory
Espoo 2008

HELSINKI UNIVERSITY OF TECHNOLOGY
Networking Laboratory
Telecommunications Engineering

ABSTRACT OF MASTER'S THESIS

Author: Carlos Muñoz Gahete

Title of thesis:
Evaluation of time synchronisation on a research network

Date: August 10 2008 **Pages:** 11 + 74
Supervisor: Raimo Kantola
Instructor: Markus Peuhkuri

During the last years, communication networks have been increasing their speed day by day. Networking research among these networks requires an accurate register of time and, furthermore, it requires the coordination of events to operate the system in unison. For that reason time synchronisation is a fundamental matter for today's communication research.

This thesis discusses several protocols and solutions improving timekeeping accuracy or providing time synchronisation. One of this solutions is based on GPS technologies that bring us important advantages such as enhanced accuracy and multi-site synchronisation.

Two different solutions providing time synchronisation through GPS technology will be studied and analysed. One of them is a homegrown card developed at Helsinki University of Technoogy and the other is a commercial off-the-self capture card. In order to carry out the performance evaluation of both devices a measurement setup has been built in the department laboratory. This measurement setup and all its components and relationships will be explained in this thesis.

Keywords: synchronisation, GPS, timing accuracy, pulse per second, SynPCI
Language: English

Acknowledgements

This work was done at the Helsinki University of Technology, at the Networking Laboratory where I have enjoyed working in the international environment provided by one of the most important universities in Europe.

I wish to thank my instructor Markus Peuhkuri for the help and the support he gave me during the execution of my thesis. I am very grateful for his suggestions, comments and guidance. I also thank him for providing me the opportunity to work in this research project.

I would like to thank my parents and my family for their endless love, encouragement and support they gave me all through my life. I also want to thank my friends from Madrid for being there when I need them and the friends I made in Finland who have been my second family during this year.

Espoo August 10th 2008

Carlos Muñoz Gahete

Abbreviations and Acronyms

CDMA	Code Division Multiple Access
CPLD	Complex Programmable Logic Device
DGPS	Differential GPS
DUCK	DAG Universal Clock Kit
FPGA	Field-Programmable Gate Array
GPS	Global Positioning System
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronic Engineering
IP	Internet Protocol
IRIG	Inter-Range Instrumentation Group
ISDN	Integrated Services Digital Network
LAN	Local Area Network
MTU	Maximum Transfer Unit
NIC	Network Interface Card
NTP	Network Time Protocol
OCXO	Oven-Controlled Crystal Oscillator
OWD	One-Way Delay
PCI	Peripheral Component Interconnect
PLL	Phase-Locked Loop
PPS	Pulse-Per-Second
PTP	Precision Time Protocol
RTT	Round Trip Time
SIM	Synchronisation Interface Module
TCP	Transfer Control Protocol
TCXO	Temperature-Compensated Crystal Oscillator
TSC	Time Stamp Counter
TTL	Transistor-Transistor Logic
UTC	Coordinated Universal Time
VCO	Voltage Controlled Oscillator
XO	Cristal Oscillator

Contents

List of Tables

List of Figures

Chapter 0

Resumen (Spanish)

0.1 Introducción

Desde el inicio de la era digital redes cada vez más rápidas han ido apareciendo día a día. Por este motivo la sincronización en red ha ido ganando importancia en el mundo de las telecomunicaciones durante los últimos 30 años. El comportamiento de la mayoría de los servicios ofrecidos por los operadores de telecomunicacines está afectado en mayor o menor medida por la calidad de su sincronización en la red.

La sincronización es un aspecto muy importante dentro de las telecomunicaciones en general, pero es aún más importante cuando nos referimos al área de investigación, dónde la precisión en el registro del tiempo y la sincronización entre los distintos equipos están directamente relacionados con los resultados de la investigación.

El uso de hardware comercial es normalmente la forma más económica de conseguir sincronización. Sin embargo, esta solución raramente se ajusta perfectamente a las necesidades de la investigación, comprometiendo así sus resultados.

En la universidad de Helsinki (HUT) se ha desarrollado un sistema de sincronización de bajo coste (tarjeta SynPCI) que permite mejorar significativamente la precisión de la sincronización, teniendo como objetivo la investigación en el área telemática.

El objetivo de este Proyecto Fin de Carrera es el diseño y construcción de una red de medida que permita la evaluación de la tarjeta SynPCI así como la de otro sistema de sincronización comercial, la tarjeta DAG de Endace, para posteriormente comparar sus prestaciones y obtener conclusiones acerca del rendimiento de ambos sistemas.

0.2 Entorno Tecnológico

Hoy en día los ordenadores utilizan distintos protocolos para conocer el tiempo correcto con una cierta precisión. Los más simples como el *daytime* y el *time protocol* proporcionan una resolución de un segundo. Para conseguir una mejor precisión existen dos protocolos: *Network Time Protocol* (NTP) diseñado para el uso en Internet, y *Precision Time Protocol* (PTP), apropiado para usarlo con dispositivos directamente conectados en una red de área local. Otros métodos estandarizados se han desarrollado para transmitir la información temporal sobre líneas dedicadas, uno de ellos es el *IRIG Serial Time Code*.

0.2.1 Fuentes de incertidumbre en las medidas de tiempo

Incluso si suponemos que el sistema operativo sabe exactamente el tiempo actual, las medidas realizadas con su reloj podrían no ser tan precisas. Vamos a describir un simple ejemplo: queremos capturar y registrar el tiempo de paquetes provenientes de una red Ethernet usando una tarjeta de red común.

Primero la tarjeta de red captura un paquete y señaliza al sistema operativo con una interrupción. El sistema operativo reacciona a la señal de interrupción y recoge el paquete de la tarjeta de red suponiendo que el bus PCI esté disponible. Si no lo está, el procesador debe esperar hasta que esté disponible. Ahora, si otro paquete ha llegado mientras el bus estaba ocupado, el procesador puede recoger ambos paquetes a la vez, haciendo parecer que ambos paquetes llegaron a la vez. Muchas tarjetas de red de altas prestaciones pueden limitar el número de interrupciones para mejorar su rendimiento, haciendo este fenómeno aún peor.

Otros factores del sistema operativo, con frecuencia provocados por el retraso de las interrupciones, añaden más imprecisiones en la medida del tiempo. Por lo tanto es necesario evaluar cuidadosamente y señalar las posibles fuentes de error en las medidas.

0.2.2 Uso de PPS (pulso por segundo) para sincronización

Una fácil solución para poder distribuir el tiempo UTC (Tiempo Universal Coordinado) entre un conjunto de ordenadores sería distribuir una señal PPS a cada ordenador utilizando cualquiera de los puertos paralelo o serie. Usando el protocolo NTP se podría obtener la numeración de los segundos y el pulso PPS señalizaría con precisión cada segundo. La ventaja de esto es su bajo coste. La construcción sería ligeramente más complicada si además queremos aislar galvánicamente el cableado entre ordenadores pero todavía por debajo de 50 euros por unidad.

Sin embargo, usar el puerto serie o paralelo para una sincronización precisa también tiene sus problemas. Para empezar los puertos tienen circuitos de protección contra descargas electrostáticas, lo que limita el tiempo de subida del reloj. Una fuente de error más importante es el hecho de que las interrupciones para estos puertos tienen prioridad baja. Esto significa que si hay alguna interrupción de mayor prioridad, como lectura/escritura en disco o cualquier comunicación por Ethernet, la interrupción será retrasada hasta que la otra haya sido atendida. Además una interrupción de alta prioridad podría reemplazar a otra de baja prioridad provocando un error en la estimación del tiempo.

0.2.3 GPS

Los protocolos de sincronización permiten a los ordenadores sincronizarse a través de la red, sin embargo son fuertemente dependientes a la carga de la red y al número de nodos entre el equipo y la fuente de tiempo. La tecnología GPS permite a cualquier equipo conseguir un *timing* perfecto con un simple receptor GPS.

El sistema GPS es un sistema de radionavegación mundial formado por una constelación de 24 satélites y sus estaciones base. Estos satélites artificiales son utilizados como punto de referencia para calcular la posiciones en la Tierra de manera muy precisa.

Sin embargo, como se detalla en el desarrollo de esta tésis, el sistema GPS no solo permite conocer la posición del receptor GPS, sino que también es posible obtener una medida muy precisa del tiempo, tan precisa como que la precisión del reloj de nuestro receptor GPS es equivalente a la de los relojes atómicos con los que están equipados los satélites.

Esto no solo permite obtener una marca de tiempo extremadamente precisa, sino que además significa que estaremos sincronizados con cualquier receptor

GPS que obtenga el tiempo del mismo modo. Si unimos a esto el hecho de que los receptores GPS son cada vez más económicos encontramos que el uso de la tecnología GPS es una solución perfecta para nuestro sistema.

0.2.4 Uso de una tarjeta adicional

Como hemos visto, existen muchas fuentes de error que afectan a la precisa sincronización del hardware. El mayor problema es la incertidumbre en el retraso indeterminado causado por las interrupciones. Si es posible reducir o eliminar esta componente, la precisión se verá mejorada de modo significativo.

Una vez más, el cambio de la prioridad de las interrupciones requeriría algunas modificaciones en el hardware por lo que no es una solución adecuada para un uso amplio. Con la señal PPS obteníamos el tiempo con precisión cada segundo, podemos cambiar esto para poder comprobar el instante de tiempo actual cuando más nos interese y suponga un mínimo retraso. Basándonos en esta información podremos hacer las correcciones adecuadas.

La idea es fabricar una tarjeta añadida (SynPCI) que disponga de dos registros. El primero de ellos será un contador que se resetee cada vez que es leído. El otro registro almacenará el valor del primer contador cuando se reciba una señal PPS. El reloj fuente para el primer contador será un oscilador controlado por tensión (VCO), enganchado en fase con una señal de 10 MHz proveniente de un reloj GPS.

Cada vez que se dispara una interrupción de tiempo, se lee el valor de 32 bits del primer registro. Este valor incluye un bit indicando si ha llegado la señal PPS y un valor de 31 bits del primer contador. El tamaño del contador es suficiente dado que es leído entre 100 y 1000 veces por segundo. El valor del kernel time-of-day xtime se actualiza de acuerdo al valor leído del registro.

El indicador PPS se usa entonces para detectar si un pulso PPS ha sido recibido entre las interrupciones del timer. Si el bit ha sido activado, entonces se lee el otro registro y se aplica una corrección a xtime si es necesario.

¿Por qué es esto mejor que PPS?

Comparado con el uso básico de la señal PPS, la principal mejora de este método es que el retraso por interrupción no tiene ningún efecto en absoluto. El tiempo entre interrupciones no puede ser mayor de 2 segundos, pero por lo demás las interrupciones pueden ocurrir en cualquier momento. Si se retrasa el acceso al bus PCI la precisión no se ve afectada ya que el valor del contador es almacenado solo cuando se lee.

0.3 Arquitectura

Para el desarrollo de la red de medida han sido utilizados los siguientes componentes del laboratorio del departamento de redes y comunicaciones:

- Generador/analizador de tráfico: Spirent Adtech AX/4000.
- Tarjeta capturadora: Endace DAG Card 3.7G.
- Tarjeta capturadora: SynPCI diseñada en la HUT.
- Receptor GPS: Trimble Acutime 2000.
- Receptor GPS: Trimble Thunderbolt GPS Disciplined Clock.
- Divisor de tráfico: VSS monitoring traffic splitter.
- Procurve Switch 1800-24G
- Tarjetas de distribución de señal.

Como dijimos en la sección anterior, el propósito de este proyecto es configurar una red de medida para ser capaces de analizar el comportamiento de dos sistemas de sincronización diferentes. Para ello, debemos conseguir una sincronización muy precisa entre los puntos críticos de la red, que son tanto el generador de tráfico como los ordenadores equipados con las tarjetas capturadoras.

La idea es mandar un patrón de tráfico desde el generador de tráfico y registrar la llegada de paquetes en los componentes que queremos analizar. Después podemos comparar estos *timestamps* con una fuente de referencia más precisa. El generador de tráfico también está equipado con un analizador de tráfico de alta calidad que podemos usar como referencia. Por esta razón se realimenta el tráfico enviado a la tarjeta capturadora de vuelta al generador de tráfico.

El generador de tráfico Spirent Adtech AX/4000 recibe una señal PPS (pulso por segundo) y una señal de 10 MHz del receptor GPS Trimble Thunderbolt a través de un cable coaxial. También recibe una comunicación serie conteniendo información temporal desde la tarjeta que estamos evaluando.

El receptor GPS Thunderbolt también proporciona la señal PPS y la señal de 10 MHz para la tarjeta capturadora SynPCI. Sin embargo en este caso la tarjeta SynPCI tiene una entrada óptica para ambas señales y por lo tanto es necesaria una conversión eléctrica-óptica.

La otra tarjeta capturadora, la tarjeta DAG de Endace, solo necesita una señal PPS para sincronizarse. Esta señal le es suministrada por el receptor GPS Trimble Acutime.

El tráfico de datos es transmitido por el generador de tráfico en formato óptico. Para introducir esta señal a los conectores Ethernet RJ45 de los equipos que contienen las tarjetas a analizar necesitamos convertirla de óptica a eléctrica. Esto se consigue pasando la señal a través de un *switch* eléctrico-óptico.

Para realimentar la señal utilizamos un divisor de tráfico, que divide la señal proveniente del generador de tráfico en dos señales. Una de ellas irá conectada a la tarjeta capturadora y la otra volverá al switch dónde será convertida de nuevo a formato óptico y realimentada al generador de tráfico.

La figura 1 muestra un esquema general con la estructura de todo el sistema.

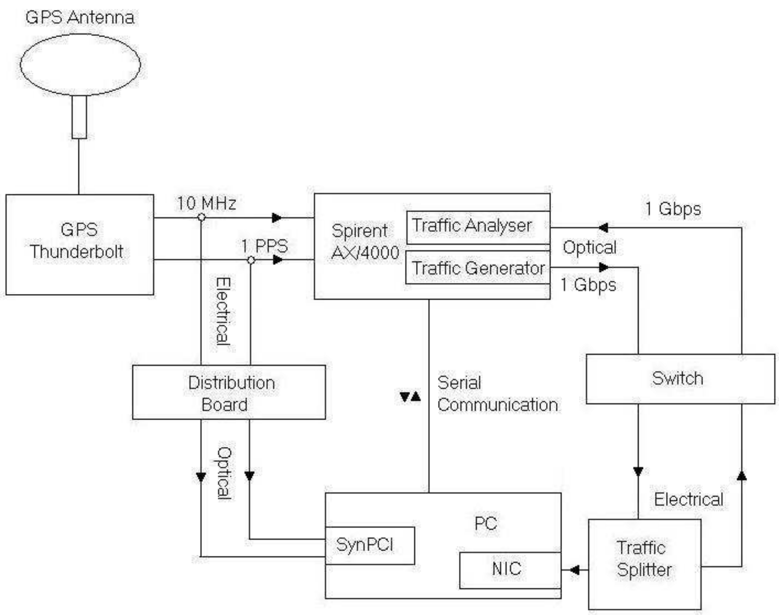

Figure 1: Diagrama del sistema de medida

0.3.1 Diseño de la tarjeta SynPCI

El diseño consiste en un chip programable CPLD (complex programmable logic device) Lattice LCMXO1200. El mismo chip proporciona funciones de bus PC y de contador. La tarjeta tiene dos entradas ópticas, una para la señal PPS y otra para la señal de 10 MHz. La frecuencia del VCO puede ser seleccionada a cualquier múltiplo de 10 MHz, hasta un máximo de 400 MHz con software. Seleccionar un correcto factor de multiplicación para el PLL hace posible contar con la resolución deseada a 2^N nanosegundos lo que resulta en una ejecución más rápida. Los componentes principales de la tarjeta SynPCI se muestran en la figura 2.

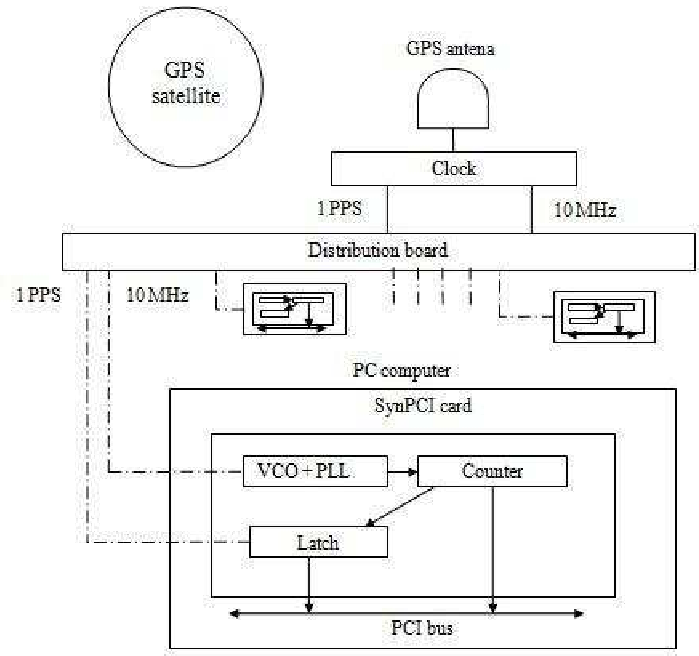

Figure 2: Componentes principales del sistema de sincronización SynPCI

El uso de señales ópticas también tiene sus ventajas. Con las señales ópticas no es necesario preocuparse sobre la referencia a tierra, y permite transmitir la señal a distancias mayores que con el uso de señales eléctricas. Los transceptores y receptores ópticos no son mucho más caros que dotar a la tarjeta de aislamiento galvánico.

La tarjeta está diseñada para poder instalarse en un armario rack. La tarjeta tiene indicadores LED para la señal de entrada, con el fin de ayudar a verificar la correcta instalación de los cables de fibra óptica. Un conector con 20 señales de entrada/salida puede usarse para distribuir señales o recibir entradas en aplicaciones futuras. Este conector y la capacidad libre del chip programable CPLD puede usarse para interconectarlo con otros sistemas.

Además de la tarjeta SynPCI, el sistema necesita también una placa de distribución que tome la señal PPS eléctrica y la señal de 10 MHz del reloj GPS y proporcione a la salida el número deseado de señales ópticas.

0.3.2 Tarjeta DAG de Endace

Para evaluar las prestaciones de la tarjeta SynPCI desarrollada en la universidad de Helsinki (HUT), vamos a comparar sus prestaciones con otros dispositivos comerciales de coste moderado. En concreto el dispositivo elegido para esta tarea será la tarjeta DAG de Endace.

La tarjeta DAG de Endace ha sido diseñada para la captura y la generación de paquetes en redes Ethernet. Los paquetes Ethernet son recogidos a través de cualquiera de las dos interfaces RJ45 de la tarjeta y transferidos a través de relays a una FPGA. Esta FPGA contiene un procesador Ethernet y el motor de sincronización que calculará los timestamps.

Dado la cercana asociación de todos sus componentes, los timestamps correspondientes a los paquetes son calculados de manera precisa. Todos estos paquetes son almacenados por la FPGA, directamente conectada con el bus PCI, y escritos en la memoria del PC. La figura 3 muestra los principales componentes y el flujo de datos de la tarjeta DAG.

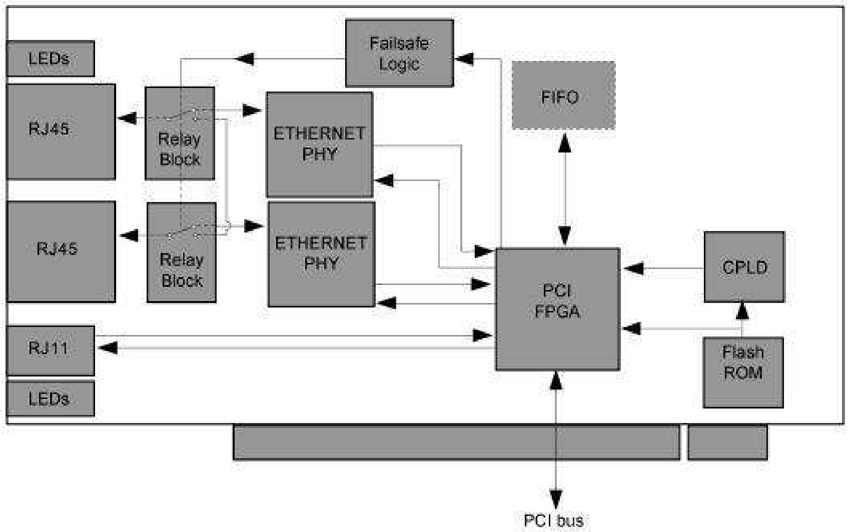

Figure 3: Estructura interna de la tarjeta DAG

0.4 Análisis

El procedimiento que seguiremos para analizar el comportamientos de las dos tarjetas capturadotas es muy simple. El tráfico de datos es enviado por el generador de tráfico a través del estándar Gigabit Ethernet, se divide antes de llegar a la tarjeta capturadora y es realimentada de vuelta al analizador de alta precisión incorporado en el generador de tráfico AX/4000. Tanto la tarjeta capturadora como el AX/4000 registraran las marcas de tiempo o '*timestamps*'.

Nuestro objetivo es medir la precisión proporcionada por las tarjetas capturadoras para monitorizar el tráfico y registrar los *timestamps* de los paquetes enviados por el generador de tráfico. Dado que no es posible medir la precisión directamente, tendremos que medir las diferencias entre los *timestamps* obtenidos de la tarjeta capturadora y del analizador del AX/4000 que será usado como referencia.

Los patrones de tráfico que hemos usado para los tests consisten en ráfagas de paquetes periódicos y de tamaño fijo. Nos interesa saber en que medida afecta la densidad del tráfico y el tamaño de los paquetes a la precisión de los *timestamps*, por ello hemos realizado diferentes tests variando el tamaño de los paquetes y la velocidad de transmisión.

La tabla 1 muestra un resumen con las estadísticas de los resultados obtenidos para los diferentes tamaños de paquete y velocidades de transmisión.

0.5 Conclusiones

La sincronización de los equipos en red es una característica muy impotante para las telecomunicaciones en general, pero es aún más importante cuando tratamos con redes de investigación experimental donde la sincronización es un aspecto fundamental si queremos obtener resultados aceptables.

Después del análisis hemos visto que el comportamiento de la tarjeta synPCI no es tan bueno como el de la tarjeta DAG, pero aún así nos ofrece una precisión mejor que la que obtendríamos conectando la señal de sincronización PPS directamente al puerto serie.

La principal ventaja de la tarjeta synPCI es que evita los retrasos provocados por las interrupciones, lo que es especialmente importante cuando el equipo se encuentra funcionando bajo altas cargas para el procesador. Otra característica importante es que el sistema se sincroniza rápidamente con la

Table 1: Resumen de los resultados obtenidos

	Mean	99^{th} perc.	interprc range	Std deviation
DAG 64 B @ 10 Mbit/s	3.303 μs	3.384 μs	185 ns	39.45 ns
DAG 64 B @ 200 Mbit/s	3.291 μs	3.356 μs	129 ns	28.39 ns
DAG 64 B @ 500 Mbit/s	3.309 μs	3.370 μs	127 ns	27.89 ns
DAG 518 B @ 10 Mbit/s	8.731 μs	8.803 μs	148 ns	32.28 ns
DAG 518 B @ 200 Mbit/s	8.753 μs	8.813 μs	120 ns	26.91 ns
DAG 518 B @ 500 Mbit/s	8.755 μs	8.816 μs	121 ns	26.71 ns
DAG 1418 B @ 10 Mbit/s	20.160 μs	20.228 μs	139 ns	30.61 ns
DAG 1418 B @ 200 Mbit/s	20.108 μs	20.170 μs	125 ns	27.81 ns
DAG 1418 B @ 500 Mbit/s	19.900 μs	19.970 μs	122 ns	27.18 ns
SynPCI 64 B @ 10 Mbit/s	14.92 μs	15.97 μs	2.08 μs	736 ns
SynPCI 64 B @ 200 Mbit/s	12.265 μs	15.85 μs	7.25 μs	2.328 μs
SynPCI 64 B @ 500 Mbit/s	N/A	N/A	N/A	N/A
SynPCI 518 B @ 10 Mbit/s	24.346 μs	25.47 μs	2.17 μs	776.35 ns
SynPCI 518 B @ 200 Mbit/s	25.547 μs	26.39 μs	1.54 μs	823.2 ns
SynPCI 518 B @ 500 Mbit/s	24.43 μs	39.30 μs	29.83 μs	10.15 μs
SynPCI 1418 B @ 10 Mbit/s	24.90 μs	25.80 μs	1.83 μs	700.7 ns
SynPCI 1418 B @ 200 Mbit/s	26.49 μs	27.35 μs	1.68 μs	821.7 ns
SynPCI 1418 B @ 500 Mbit/s	27.91 μs	28.72 μs	1.42 μs	899.1 ns

fuente GPS tras ser reiniciado. La entrada por fibra óptica provee a la tarjeta synPCI con inmunidad contra ruidos electromagnéticos y una atenuación menor. Sin embargo su problema más importante es la degradación de su comportamiento cuando trabaja con gran densidad de tráfico.

La tarjeta DAG ha demostrado tener un mejor rendimiento que la tarjeta synPCI. Es capaz de obtener timestamps más precisos y su comportamiento no se ve afectado por la densidad de tráfico recibida en sus puertos de entrada. Apesar de ser una tarjeta comercial, es algo más cara que la tarjeta synPCI. El tiempo que requiere esta tarjeta para sincronizarse con la señal PPS es otra desventaja de la tarjeta DAG respecto a la synPCI.

En resumen, en esta tésis hemos estudiado tecnologías de sincronización que nos ayudan a mantener un registro del tiempo más preciso, capacidad de sincronización en múltiples localizaciones, y todo esto a un precio razonable que permite la escalabilidad del sistema. Una mejor precisión en los resultados permitirá que la investigación se centre más en el objetivo de la misma que en los posibles errores obtenidos de una mala sincronización entre equipos.

Chapter 1

Introduction

In digital systems, many operations must follow a precedence relationship. If some operations follow a rule of precedence, then synchronisation ensures that operations run in the right order. At the hardware level, synchronisation is accomplished by distributing a common timing signal to all the modules of the system. At a higher level of abstraction, software processes synchronise by exchanging messages.

1.1 Synchronisation in Telecommunications

Since the beginning of the Internet era, faster and faster networks have been coming out day by day. For that reason, network synchronisation has gained increasing importance in telecommunications throughout the last 30 years. The quality of most services offered by telecommunications operators is affected directly by their network synchronisation performance, especially since transmission and switching became digital.

Lack of synchronisation in digital switching equipments can cause timing slips degrading the system performance. Even if normal telephone conversation are not significantly affected, switched data services are more problematics dealing with synchronisation slips. Therefore, on circuit-switched data networks like ISDN, synchronisation becomes a fundamental matter for the proper behaviour of the network.[14]

1.2 Synchronisation for Networking Research

If synchronisation is a very important issue in telecommunications in general, it becomes even more important when we refer to the research area, where correct timekeeping is related directly with the results of the research. Although sometimes a coarse accuracy for timekeeping is enough, experimental research among computers connected by the fastest networks often requires a more accurate record of time.

Furthermore, if the experiment needs timekeeping in more than one device, a very precise accuracy is usually not enough and it often requires the coordination of events to operate the system correctly. That makes the synchronisation of the clocks in the network a major issue in order to reach the research goals and avoid error due to differences between clocks.

Commercial off-the-shelf hardware is usually the most economic way to achieve synchronisation. Cheap synchronisation devices provide an easy and scalable solution, however it is seldom the most suitable alternative, compromising the results of the research in terms of accuracy and synchronisation.

Some commercials systems can provide very accurate timestamping and synchronisation. However, this is a very high cost solution and in most of the cases unaffordable for research institutions, specially when synchronisation is required in a wide research network.

1.3 Problem Statement

As we said, off-the-shelf products provide an economic solution but are rarely the best option. At Helsinki University of Technology, an inexpensive synchronisation system has been developed trying to improve significantly timestamping accuracy.

The system has been designed with the aim of networking research. This research includes networking experimentation using different algorithms and protocols. If we want to analyse the difference between these algorithm and protocols, at least microsecond resolution is required. In order to obtain reliable results, devices related to traffic generation and analysis are critical points where synchronisation is of paramount importance.

Networking experimentation usually requires to repeat the same traffic patterns over and over again. Without a proper synchronisation among traffic generators and analysis points, the researcher would not be able to distinguish significant differences in the behaviour of algorithms from the noise generated by synchronisation errors.

This thesis discusses different protocols and systems designed to improve timekeeping accuracy and synchronisation, analysing their advantages and limitations. Finally it presents the solution given at Helsinki University of Technology consisting of one add-on card that provides accurate timestamping by avoiding interrupts latency.

1.4 Goal of the Thesis

The goal of the thesis is to feed the traffic generator and the capture cards with GPS signals in order to achieve synchronisation among devices and build a measurement setup to collect the data. The final task will be to analyse and evaluate the performance of two different systems in terms of synchronisation accuracy. One of the key aspects is to identify possible sources of error and distribution of error.

Chapter 2

Background

In this chapter we will give an overview on how common computers manage to keep their clocks synchronised and to control the timing of actions. We will also introduce some other protocols and methods to improve this basic synchronisation.

2.1 Basis and Technology of Clocks

From a theoretical point of view, the operation principle of any kind of clock consists of a generator of oscillations and an automatic counter of such oscillations. With different ability, the oscillator can be based indeed on (pseudo-)periodic physical phenomena of any kind: the swinging of a pendulum or a wheel in mechanical clocks, the vibration of atoms in a crystal around their minimum energy position in quartz clocks, the radiation associated with specific quantum atomic transition in atomic clocks are just the best known examples due to their widest application but not the only ones.

2.1.1 Quartz-Crystal Clocks

Quartz-crystal oscillators are based on piezoelectric effect: a mechanical strain in the crystal yields an electrical field and vice versa. A Crystal Oscillator (XO) is thus an electronic oscillator, where a quartz crystal is excited by a periodic electrical signal at the resonance frequency (in the range from 10 kHz to 1 GHz, but most commonly from 5 to 10 MHz).

The resonance frequency is determined mainly by the properties of the bulk material such as size, shape, elasticity, and the speed of sound in the material, but it is also strongly dependent on environmental conditions such as the temperature, humidity, etc. (for example, temperature affects crystal dimensions). Used in a positive feedback circuit, it allows to generate a timing signal featuring an excellent short-term stability (in particular over observation intervals smaller than one second).

A crystal's frequency characteristic depends on the shape or 'cut' of the crystal. High-frequency crystals are typically cut in the shape of a simple, rectangular plate. Low-frequency crystals, such as those used in digital watches, are typically cut in the shape of a tuning fork. A tuning fork crystal is usually cut such that its frequency over temperature is a parabolic curve centred around 25 ℃. This means that a tuning fork crystal oscillator will resonate close to its target frequency at room temperature, but will slow down when the temperature either increases or decreases from room temperature. A common parabolic coefficient for a 32 kHz tuning fork crystal is -0.04 ppm/℃^{2}.

$$f = f_0[1 - 0.04ppm(T - T_0)^2] \tag{2.1}$$

For the usually used quartz crystal cuts, their frequency-temperature characteristics are shown in Figure 2.1.

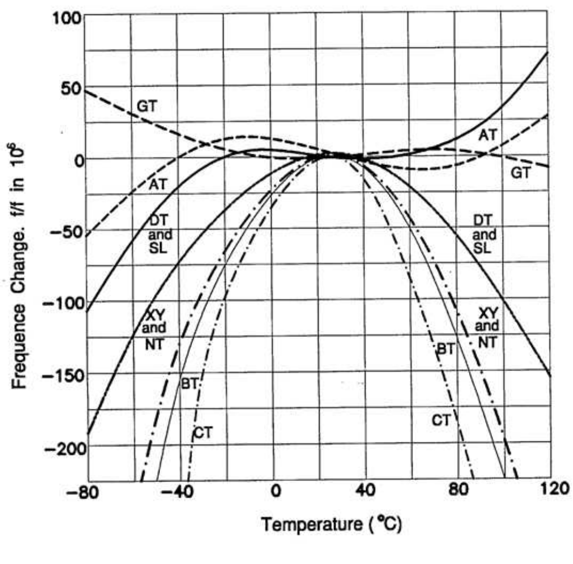

Figure 2.1: Frequency-temperature characteristics of various quartz cuts.

2.1.2 Temperature-Compensated Crystal Oscillator

The main problem of a plain XO is the dependence of its natural frequency on ageing (around 10^{-7}/day in plain models) and on the temperature (in the order of 10^{-7}/°C or above).

To overcome the latter problem, Temperature-Compensated Crystal Oscillators (TCXOs) implement an automatic control on the oscillation frequency based on the measurement of the crystal temperature. Such a trick allows to achieve a frequency stability of 10^{-7} over a temperature interval from 0°C to 50°C. More sophisticated models by way of digital control, achieve a frequency stability even in the order of 10^{-8} in the wider temperature interval from 0°C to 70°C.

2.1.3 Oven-Controlled Crystal Oscillator

Far better than compensating temperature variations with a feedback control is to insulate the oscillator thermically and to make it work in a constant-temperature closed environment. Such clocks are called Oven-Controlled Crystal Oscillators (OCXOs).

In OCXOs, the resonator and the other temperature-sensitive elements are placed in a controlled oven whose temperature is set as closely as possible to a point where the resonator frequency does not depend on temperature, so to minimise the effect of residual temperature variations. Frequency stability values exceeding 10^{-9}/day are thus achieved. [7] [9] [15]

2.1.4 Crystal Oscillators in Computers

Common PC computers use quartz-crystal oscillators integrated on the motherboard. Quartz oscillators allow computers to keep a register of time, using different timers such as Time Stamp Counter (TSC) that provides high-resolution timestamps. However, these oscillators are directly influenced by the heat dissipation coming from the motherboard and, although their accuracy is stable in the short-term, changes of temperature destabilise accuracy in the long-term.

One could think that improving stability is easily achieved using TCXO or OCXO instead of common oscillators. The problem is that these oscillators are not usually found embedded on common motherboards and it would need a specific design for the mother board what takes not only time but money. Furthermore this solution does not help for time synchronisation among different computers, so we will need another solution that implies synchronisation over the network. [11]

2.2 Synchronisation over the Network

In order to be in sync, computers exchange messages over the network. With that purpose, computers use different protocols and services. Some of them just provide us with a resolution of one second like daytime protocol. In this section we present other more accurate protocols: Network Time Protocol (NTP), Precision Time Protocol (PTP), and the IRIG Serial Time Codes.

2.2.1 Network Time Protocol

The Network Time Protocol is a protocol for synchronising the clocks of computer systems over packet-switched, variable-latency data networks. NTP uses UDP as its transport layer. It is designed particularly to resist the effects of variable latency (jitter).

NTP has a tree structure as shown in Figure 2.2. Stratum 1 computers are attached to reference clocks , typically GPS clocks or atom clocks . Normally they act as servers for timing requests from stratum 2 servers via NTP. And thus, every stratum works as a server for the next stratum.

When a computer wants to get synchronisation using NTP, it exchanges packages containing timestamps and then makes a correction of time using those timestamps. NTP protocol usually provides an accuracy of few milliseconds on a unloaded network but it is highly dependant on the network behaviour and the distance with the stratum 1 servers.

2.2.2 Precision Time Protocol

Network Time Protocol (NTP) targets large distributed computing systems with millisecond synchronisation requirements. However, it does not fulfil the requirements for smaller networks when a better accuracy is needed.

The Precision Time Protocol (PTP) is a time-transfer protocol defined in the IEEE 1588 standard. This protocol is applicable to distributed systems consisting of one or more nodes, communicating over some set of communication media. Nodes are modelled as containing a real-time clock that may be

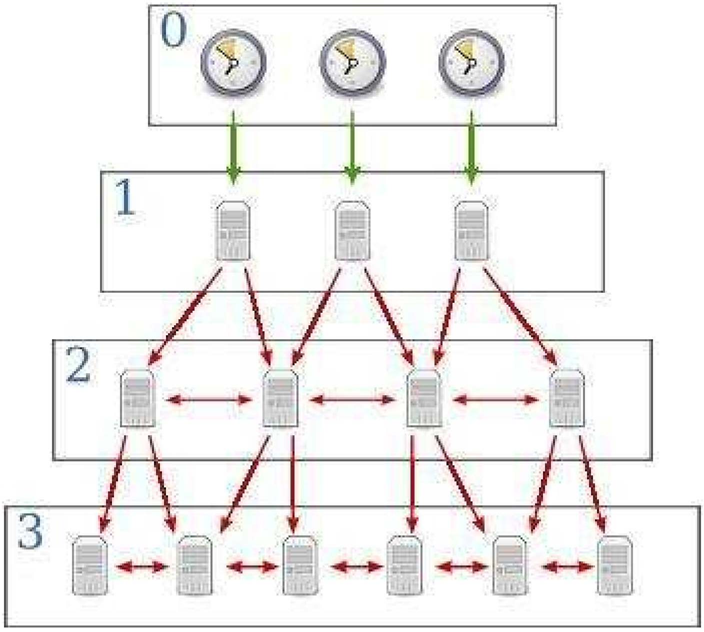

Figure 2.2: Structure of NTP servers [3]

used by applications within the node for various purposes, such as generating timestamps for data or ordering events managed by the node. The PTP protocol provides a mechanism for synchronising the clocks of participating nodes to a high degree of accuracy.

In the protocol, the master device periodically launches an exchange of messages with slave devices to help each slave clock recompute the offset between its clock and the master's clock. This offset will drift with time, and so these periodic exchanges mitigate the impact of this drift on clock synchronisation. One assumption is that this exchange of messages happens over a period of time so small that this offset can safely be considered constant. Another assumption is that the transit time of a message going from the master to a slave is equal to the transit time of a message going from the slave to the master. Finally, it is assumed that both the master and slave can measure the time they send or receive a message. The degree to which these assumptions are enforced in an application regulate the accuracy of the offset measured at a slave device.

The sequence of packets used to transfer time from the PTP master to a PTP slave can be visualised in Figure 2.3. Sync packets are stamped as they leave the master and arrive at the slave. The difference in the master and slave time-stamps represents the offset of the slave plus the message transmission delay. The slave clock adjusts itself to compensate for this difference. To correct for message-transmission delay, the slave uses a second set of sync and follow-up messages with its corrected clock to calculate the master-slave delay. The second set of messages accounts for variations in network delays. The slave time-stamps and sends a delay request message. The master time-stamps the arrival of this message and sends a delay response message with the delay request arrival time-stamp. The difference between the two time stamps is the slave-to-master delay. The slave averages, the two directional delays and then adjusts its clock by the delay to synchronise the two clocks. The sequence repeats periodically to keep the two clocks synchronised.[6]

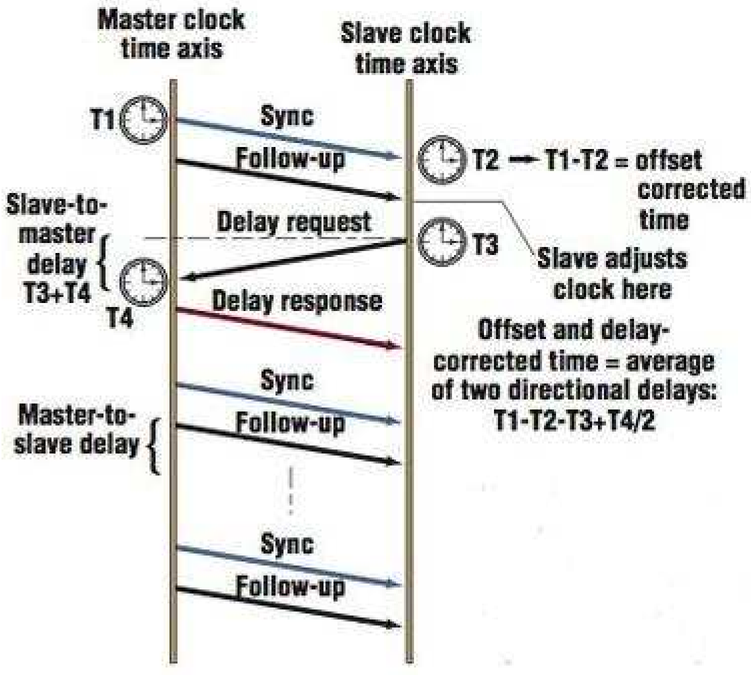

Figure 2.3: PTP Sequence of Packets [1]

2.2.3 IRIG Serial Time Code

Inter-range instrumentation group time codes, commonly known as "IRIG" time codes, were created by the TeleCommunications Working Group of the Inter-Range Instrumentation Group, the standards body of the Range Commanders Council. The latest version of the Standard is IRIG Standard 200-04. This protocol allows to send current time code through an electrical or optical transmission. The different time codes defined in the Standard have alphabetic designations. Table 2.1 shows the standards currently defined. The main difference between codes is their rate, which varies between one pulse per minute and 10.000 pulses per second.

Table 2.1: Pulse Rates of IRIG time code formats[13]

Format	Pulse Rate	Time Interval
A	1 kpps	1 millisecond
B	100 pps	10 millisecond
D	1 ppm	1 minute
E	10 pps	0.1 second
G	10 kpps	0.1 millisecond
H	1 pps	1 second

2.2.4 Use of Pulse-Per-Second

A Pulse Per Second (PPS) is an electrical signal that very precisely indicates the start of a second. PPS signals are output by various types of precision clock. Depending on the source, properly operating PPS signals have an accuracy ranging from a few nanoseconds to a few milliseconds. Note that because the PPS signal does not specify the time but merely the start of a second, one must combine the PPS functionality with another time source that provides the full date and time, in order to ascertain the time both accurately and precisely.

PPS signals are used for precise timekeeping and time measurement. One increasingly common use is in computer timekeeping, using it together with the NTP protocol. PPS provides us a cheap solution for time synchronisation, it is easily distributed to all computers in the network and it does not need a complicated construction. For our purposes we will use a GPS receiver as a source for PPS signalling, so we can get a high accurate timing from the PPS signal supported by the short-term stability provided by the crystal oscillator. GPS technology is explained in the next section.

2.3 GPS

Synchronisation protocols allow computers to be in sync over the network, but are strongly dependant on the network load and on the number of nodes between the computer and the time source. In this section we explain the basis of GPS and how we can achieve perfect timing with a simple GPS receiver.

The Global Positioning System (GPS) is a worldwide radio-navigation system formed from a constellation of 24 satellites and their ground stations. GPS uses these artificial satellites as reference points to calculate positions accurate to a matter of meters. In fact, with advanced forms of GPS you can make measurements to better than a centimetre. GPS receivers have been miniaturised to just a few integrated circuits and so are becoming very economical, what makes the technology accessible to virtually everyone.

The GPS system can be devided into three basic segments which will be discussed below:

- Space segment (satellites).
- Control segment (control stations).
- User segment (GPS receiver).

2.3.1 Space Segment

The space segment consists of 24 satellites orbiting in 6 different orbits as we can see in figure 2.4. The first of the satellites was brought to its orbit as early as 1978. During the years the satellites became more and more sophisticated and meanwhile five different types of these satellites exist (Block I, Block II, Block IIA, Block IIR und Block IIF).

2.3.2 Control Segment

The GPS-System is controlled by the US Army. The "master control station" (Schriever AFB) and four additional monitoring stations (on Hawaii, Ascension Islands, Diego Garcia and Kawajalein) were set up for monitoring the satellites. During August and September 2005, six more monitor stations of the NGA (National Geospatial-Intelligence Agency) were added to the grid. Now, every satellite can be seen from at least two monitor stations. This allows to calculate more precise orbits and ephemeris data. For the end user, a better position precision can be expected from this. In the near future, five more NGA stations will be added so that every satellite can be seen by at least three monitor stations. This improves integrity monitoring of the satellites and thus the whole system.

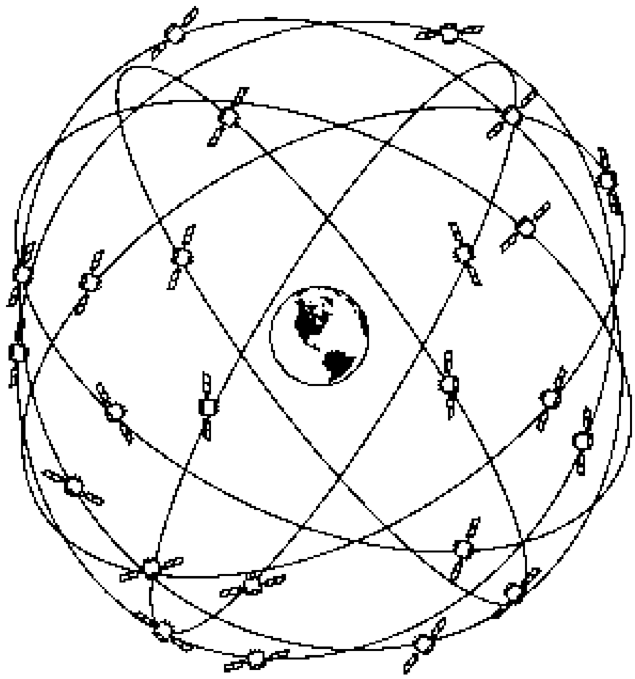

Figure 2.4: GPS satellites constellation and orbits

2.3.3 User Segment (GPS Receiver)

Modern GPS satellite receivers can be built in such a compact way, that they even can be integrated in wrist watches. Most of the commercially available instruments today have about the size of a cell phone. All receivers today have at least 12 channels, meaning that they can receive and process the signals of at least 12 satellites in parallel. Earlier instruments had to process data in series, making them considerably slower and less accurate and more sensitive against disturbances. Instruments for the professional use (military and land survey) typically are bigger and considerably more precise.

In general, GPS receivers are composed of an antenna, tuned to the frequencies transmitted by the satellites, receiver-processors, and a highly-stable clock (often a crystal oscillator). Although the GPS receiver can get perfect timing from the satellite signals as we will see later, it needs a local crystal oscillator to provide the necessary stable and accurate internal frequency standard.

2.3.4 How GPS Works

Basically, GPS works following these five steps:

1. The idea behind GPS is "triangulation" from satellites.
2. In order to triangulate, a GPS receiver calculates distance using the travel time of satellite signals.
3. To measure travel time, GPS needs very accurate timing which is achieved with some tricks we will explain later.
4. Furthermore, you need to know the exact position of satellites in space. This can be achieved basing on high orbits and satellite monitoring.
5. Finally there are some sources of errors you must correct due to delays the signal experiences as it travels through the atmosphere.

2.3.5 Triangulation

The basis of GPS is to use satellites in space as reference points for locations here on earth. By very accurately measuring our distance from three satellites we can "triangulate" our position anywhere on earth. We will explain this with an example:

1. Suppose we measure our distance from a satellite and find it to be 18.000 meters. Knowing that we are 18.000 meters from a particular satellite narrows down all the possible locations we could be in the whole universe to the surface of a sphere that is centred on this satellite and has a radius of 18.000 miles.
2. Next, say we measure our distance to a second satellite and find out that it is 19.000 meters away. That tells us that we are not only on the first sphere but we are also on a sphere that is 19.000 meters from the second satellite. Or in other words, we are somewhere on the circle where these two spheres intersect.
3. If we then make a measurement from a third satellite and find that we are 20.000 meters from that one, that narrows our position down even further, to the two points where the 20.000 meters sphere cuts through the circle that is the intersection of the first two spheres.

> So by ranging from three satellites we can narrow our position to just two points in space. To decide which one is our true location we could make a fourth measurement. But usually one of the two points is a ridiculous answer (either too far from Earth or moving at an impossible velocity) and can be rejected without a measurement.

2.3.6 Measuring Distance

We saw in the last section that a position is calculated from distance measurements to at least three satellites. Distance can be easily calculated multiplying the signal travel time by the speed of light. The problem is measuring the travel time.

The timing problem is tricky. First, the times are going to be extremely short. If a satellite were right overhead the travel time would be something like 0.06 seconds. So we are going to need some really precise clocks.

To make the measurement we assume that both the satellite and our receiver are generating the same pseudo-random codes at exactly the same time. By comparing how late the pseudo-random code of the satellite appears compared to the code of the receiver, we determine how long it took to reach us.

Thus, if we designate the coordinates and the time sent as $[x_i, y_i, z_i, t_i]$ where i is the indicator for each one of the 4 satellites, and knowing the arrival time at the receiver tr_i, we can calculate the transit time of the message, $(tr_i - t_i)$. Assuming the transmission speed is the speed of the light, c, the distance travelled by the message is

$$p_i = (tr_i - t_i)c \tag{2.2}$$

Once we know the position of the satellites we can triangulate the position of the receiver as we mentioned previously.

2.3.7 Getting Perfect Timing

At the speed of light a timing error of just a thousandth of a second is translated into 300 kilometres of error. For that reason we need very accurate and synchronised clocks in both sides.

On the satellite side, timing is almost perfect because they have highly precise atomic clocks on board. But on the receivers, atomic clocks are just unaffordable.

Luckily GPS allows us to get by accurate clocks in our receivers with a brilliant little trick. This trick is one of the key elements of GPS and as an added side benefit it means that every GPS receiver is essentially an atomic-accuracy clock.

The secret to perfect timing is to make an extra satellite measurement. If three perfect measurements can locate a point in 3-dimensional space, then four imperfect measurements can do the same thing.

If our receiver's clocks were perfect, then all our satellite ranges would intersect at a single point (which is our position). But with imperfect clocks, a fourth measurement, done as a cross-check, will not intersect with the first three. That makes the receiver realises that its clock it is not perfectly synchronised with universal time.

Since any offset from universal time will affect all of our measurements, the receiver looks for a single correction factor that it can subtract from all its timing measurements that would cause them all to intersect at a single point.

That correction brings the receiver's clock back into synchronisation with universal time, and that is how we get an atomic accuracy clock from a simple receiver.

Mathematically, if b denotes the clock error or bias, we have four unknowns, the three coordinates of the GPS receiver and the clock bias $[x, y, z, b]$, and four equations corresponding to the spheres of the satellites given by:

$$(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 = ((tr_i + b - t_i)c)^2, i = 1, 2, 3, 4. \tag{2.3}$$

Another way to express these equations is in terms of pseudo-ranges, which are approximations to the distance travelled by the message as in equation 2.2 but having in account the bias unknown. [8]

$$p_i = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} - bc, i = 1, 2, 3, 4. \tag{2.4}$$

To solve these equations there exists different methods. The most important is trilateration followed by one dimensional root finding and multidimensional Newton-Raphson calculations. [12]

2.3.8 Error Sources

Up to now we have considered our GPS system very abstractly and we have not introduced the effects produced by the different sources of error and interferences.

Multipath effect

The multipath effect is caused by reflection of satellite signals (radio waves) on objects. It was the same effect that caused ghost images on television when antennas on the roof were still more common instead of today's satellite dishes.

For GPS signals this effect mainly appears in the neighbourhood of large buildings or other elevations. The reflected signal takes more time to reach the receiver than the direct signal. The resulting error typically lies in the range of a few meters.

Clock Inaccuracies and Rounding Errors

Despite the synchronization of the receiver clock with the satellite time during the position determination, the remaining inaccuracy of the time still leads to an error of about 2 m in the position determination. Rounding and calculation errors of the receiver sum up approximately to 1 m.

Relativistic Effects

The following section shall not provide a comprehensive explanation of the theory of relativity. In the normal life we are quite unaware of the omnipresence of the theory of relativity. However, it has an influence on many processes, among them is the proper functioning of the GPS system. This influence will be explained shortly in the following.

As we already learnt, the time is a relevant factor in GPS navigation and must be accurate to 20 - 30 nanoseconds to ensure the necessary accuracy. Therefore the fast movement of the satellites themselves (nearly 12000 km/h) must be considered.

Whoever already dealt with the theory of relativity knows that time runs slower during very fast movements. For satellites moving with a speed of 3874 m/s, clocks run slower when viewed from earth. This relativistic time dilation leads to an inaccuracy of time of approximately 7.2 microseconds per day.

The theory of relativity also says that time moves the slower the stronger the field of gravitation is. For an observer on the earth surface the clock on board of a satellite is running faster (as the satellite in 20000 km height is exposed to a much weaker field of gravitation than the observer). And this second effect is six times stronger than the time dilation explained above.

Altogether, the clocks of the satellites seem to run a little faster. The shift of time to the observer on earth would be about 38 milliseconds per day and would make up for an total error of approximately 10 km per day. In order that those error do not have to be corrected constantly, the clocks of the satellites were set to 10.229999995453 MHz instead of 10.23 MHz but they are operated as if they had 10.23 MHz. By this trick the relativistic effects are compensated once and for all.

Atmospheric effects

Another source of inaccuracy is the reduced speed of propagation in the troposphere and ionosphere. While radio signals travel with the velocity of light in the outer space, their propagation in the ionosphere and troposphere is slower.

In the ionosphere in a height of 80 - 400 km a large number of electrons and positive charged ions are formed by the ionising force of the sun. The electrons and ions are concentrated in four conductive layers in the ionosphere (D, E, F1, and F2 layer). These layers refract the electromagnetic waves from the satellites, resulting in an elongated run time of the signals.

There are a couple of ways to minimise this kind of error. For one thing we can predict what a typical delay might be on a typical day. This is called modelling and it helps but, of course, atmospheric conditions are rarely exactly typical.

Another way to get a handle on these atmosphere-induced errors is to compare the relative speeds of two different signals. This "dual frequency" measurement is very sophisticated and is only possible with advanced receivers.

Other source of error is due to reflections of the signal in local obstructions before it gets to the receiver. This is called multipath error. Good receivers use sophisticated signal rejection techniques to minimise this problem.

Fortunately all of these inaccuracies still do not add up to much of an error. And a form of GPS called "Differential GPS" can significantly reduce these problems.

2.3.9 Differential GPS

Differential GPS (DGPS) involves the cooperation of two receivers, one that is stationary and another that is roving around making position measurements. The stationary receiver is the key. It ties all the satellite measurements into a solid local reference.

Since each of the timing signals that go into a position calculation has some error, that calculation is going to be a compounding of those errors. Luckily the satellites are located in high orbits far away in space so the little distances we travel here on earth are insignificant.

So if two receivers are fairly close to each other, say within a few hundred kilometres, the signals that reach both of them will have travelled through virtually the same path in the atmosphere, and so will have virtually the same errors.

That is the idea behind differential GPS: We have one receiver measure the timing errors and then provide correction information to the other receivers that are roving around. That way virtually all errors can be eliminated from the system.

The idea is simple. Put the reference receiver on a point that has been very accurately surveyed and keep it there. This reference station receives the same GPS signals as the roving receiver but instead of working like a normal GPS receiver it attacks the equations backwards.

Instead of using timing signals to calculate its position, it uses its known position to calculate timing. It figures out what the travel time of the GPS signals should be, and compares it with what they actually are. The difference is an "error correction" factor.

Since the reference receiver has no way of knowing which of the many available satellites a roving receiver might be using to calculate its position, the reference receiver quickly runs through all the visible satellites and computes each of their errors. Then it encodes this information into a standard format and transmits it to the roving receivers. The roving receivers get the complete list of errors and apply the corrections for the particular satellites they are using. [2] [5] [10]

2.4 Error Sources in Computer's Time Keeping

At this moment we have achieved to feed all the devices in our network with an atomic precision clock, but even with that feed we still have some other uncertainty sources to deal with.

Since the network interface card (NIC) captures a packet until the packet is time stamped, the system needs to pass through a few steps. Figure 2.5 shows a schematic of the timestamping process. The packet arrives at time t_2, which is the real time of arrival, after that, the NIC sends an interrupt signal to the operating system, then the operative system realises the new packet and tries to recover the packet from the NIC. However, if the PCI bus is not available the system has to wait until it becomes available again causing a delay in the time stamp process and thus, decreasing the accuracy of the time stamp, which occurs at time t_2'. Furthermore, if another packet arrives while the buffer is busy, the system can process both packets together and they will get the same time stamp.

When sending the packet there is also an inaccuracy in the timestamp due to the timestamp is set by the application at time t_1', but the packet is not really sent by the NIC until time t_1.

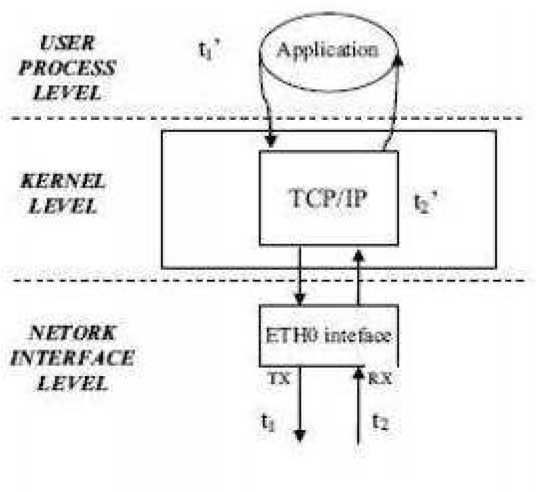

Figure 2.5: Timestamping process

There are also another factors that contribute to timing inaccuracies. Most of them are induced by interrupt latency. The main problem is that these factor are seldom directly measurable so we need to develop a measure system to be able to detect the error sources and afterwards try to correct them. [11]

2.5 Two-Way and One-Way Delay

The two-way delay of packets is an interesting characteristic of Internet. It provides extensive information about state of networks and application performance. For network characterisation, we can get values of transmission and propagation delays looking at the minimum two-way delay. The variations of delay are related with queueing effects. Even we can infer topology information, congestion state or route changes from two-way delay measurements. From an application performance point of view, large delays will make difficult to obtain a sustainable high bandwidth flow because of TCP dependency on RTT (Round Trip Time).

Two-way delay can be divided into several components:

- Transmission delay: the time needed to put all bits of a packet over a data link.
- Propagation delay: the time needed to propagate a bit though the data link.
- Processing delay: the time needed to process a packet in each router.
- Queueing delay: the time needed to wait in a router queue before transmission.

Processing and queueing delays are stochastic random variables due to variability in processing tasks in routers and in network conditions respectively. However, nowadays processing time should be almost constant except on rare occasions because modern IP routers use hardware assisted forwarding with wire speed. Transmission and propagation delays are almost constant for a certain path, because they depend on link capacity and distance respectively. Propagation and transmission delays will give a good approximation to the minimum two-way delay and this minimum will have to be a constant value.

Normally, two-way delay is approximated by RTT. Those measurements are very easy to obtain, using ICMP Request/Reply packets through the ping tool and controlling only one of the end nodes. However, asymmetry of paths makes this approximation invalid because delay can be different for both directions of a path. Actually, for two way delay measurement, it does not matter that routes are asymmetrical. One can not divide the RTT by 2 and assume that the result is a one way delay if routes are asymmetrical.

Another procedure is measuring one-way delay (OWD) which is a more complex measurement. It requires expensive hardware but it provides a better characterisation of the parameter.[4]

Two-way delay does not need synchronisation since all the measures are done in the same point of the network. However, OWD does need synchronisation and it is the measure we will use in this thesis to analyse the performance of the different devices.

Chapter 3

Architecture

This chapter shows in detail how is the architecture of the measurement system and provides a description of the testing environment. We will also present all the different devices included in the system and their characteristics.

3.1 Environment

For the development of the measurement setup we have used the following components of the Department of Communications and Networks research laboratory:

- Traffic generator: Spirent Adtech AX/4000.
- Capture card: Endace DAG Card 3.7G.
- Capture card: SynPCI developed in TKK.
- GPS receiver: Trimble Acutime 2000.
- GPS receiver: Trimble Thunderbolt GPS Disciplined Clock.
- VSS monitoring traffic splitter.
- Procurve Switch 1800-24G
- Distribution boards.

3.2 System Overview

As we said in section 1.4 the purpose of this thesis is to build a measurement setup to be able to analyse the performance of two different synchronisation systems. In order to do that, we have to achieve accurate synchronisation among the critical points of the network which are the traffic generator and the PCs equipped with the capture cards.

The idea is to send a traffic pattern from a traffic generator and timestamp the incoming packets in the devices we want to test. Then we can compare this timestamps with a more accurate reference. To get this reference we use the same traffic generator which is also equipped with a high-quality traffic analyser. For that reason we loop the traffic sent to the capture card back to the traffic generator.

The Spirent Adtech AX/4000 traffic generator receives a PPS signal and a 10 MHz signal from the Trimble Thunderbolt GPS receiver trough a coaxial cable line. It also receives a serial input providing time information from the device under test as we will see later in section 3.3.5.

Thunderbolt GPS receiver also provides both PPS and 10 MHz signals for the SynPCI capture card. However, in this case the SynPCI card is expecting optical inputs and therefore a electrical/optical conversion is needed. This conversion is performed in a distribution board that interfaces the GPS receiver and the SynPCI card.

The other capture card, Endace DAG card, only needs a PPS signal for synchronisation. It gets it from the Trimble Acutime 2000 GPS receiver.

The data transmission is output by the traffic generator in optical format. In order to input to the RJ45 Ethernet connectors of the PCs containing the cards we want to analyse, we need to convert this signal from optical to electrical. We achieve that using an optical/electrical switch.

To loop back the signal we use a traffic splitter that divides the signal into two signals. One of them goes to the capture card and the other returns to the switch, is converted again to optical and is fed back to the traffic generator.

Figure 3.1 shows the general scheme of the whole system and how it is structured.

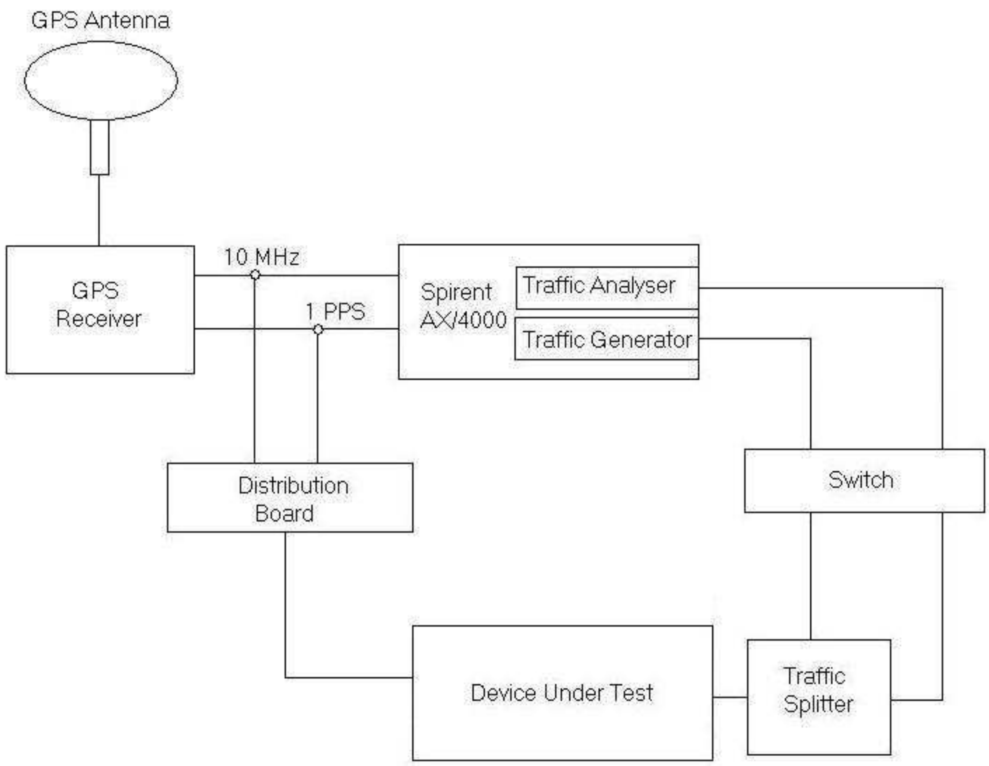

Figure 3.1: Measurement system diagram

3.3 System Components

This section explains the characteristics of all the devices and components of the measurement system and how they are interfaced with the rest of the components.

3.3.1 GPS Receivers

Trimble Thunderbolt and Trimble Acutime GPS clocks provide the system with the timing signals required to synchronise the system. The reason to use two different GPS receivers is just the availability in the laboratory and the ease to interface with the other devices.

Trimble Thunderbolt GPS Disciplined Clock

The Thunderbolt clock provides a PPS signal and a 10 MHz signal needed by the traffic generator and the SynPCI card. Both signals are electrical and are delivered directly to the traffic generator using a BNC connector and through a distribution board to the SynPCI card. Table 3.1 shows its performance specifications.

Table 3.1: Trimble Thunderbolt Performance Specifications

Pulse width	10 microseconds
PPS Accuracy (one sigma)	20 nanoseconds
10 MHz Accuracy	1.16 x 10^{-12} (one day average)

Trimble Acutime 2000

Trimble Acutime GPS Antenna is interfaced with the system using the Synchronisation Interface Module (SIM). The antenna sends the PPS signal with RS-422 signalling levels, the SIM outputs the same signal with TTL level. It is connected to the distribution board with a BNC connector. Table 3.2 shows its performance specifications.

Table 3.2: Trimble Acutime Performance Specifications

Pulse width	10 microseconds
PPS Resolution	80 nanoseconds
PPS Accuracy (one sigma)	50 nanoseconds

3.3.2 Distribution Boards

The distribution boards included in the architecture of the system have two different purposes. First of all, they provide us with the capability of sharing the GPS signals with other devices in the laboratory, and, in the case of the SynPCI card, the distribution board also fulfils the electrical/optical conversion of both PPS and 10 MHz signal.

3.3.3 Procurve Switch 1800-24G

As we previously said, the switch is used just as a optical/electrical and viceversa converter. The Procurve Switch 1800-24G is a web-managed switch with 24 ports. Two of them are dual ports that can work either with a RJ45 interface or a mini-GBIC optical interface. A latency less than 3 μs is guaranteed by the manufacturer for a Gigabit Ethernet connection and 64 bytes packets.

3.3.4 Traffic Splitter

In order to split the traffic, we will use a the VSS monitoring 10/100/1000 1x1 Copper Tap. It allows the uninterrupted pass through of full duplex data over its two RJ45 traffic ports, while the network signals are duplicated to the transmit-only monitoring ports.

Decoding the network signals this device reclocks the data and electronically replicates an exact copy (including line errors) to two transmit-only RJ45 monitoring ports, thereby giving the user the ability to monitor a 10/100/1000 copper link with a gigabit copper monitoring device.

The traffic generator sends the packets to one of the traffic ports and this signal is looped back through the other traffic port. The duplicated signal is output by one of the monitoring ports and connected to capture card we want to test.

3.3.5 Traffic Generator

The traffic generator that we will use to send the packets to the devices we want to test is the Spirent Adtech AX/4000. AX/4000 is a modular, multi-port system capable of testing multiple transmission technologies such as ATM, IP, Frame Relay and Ethernet simultaneously at speeds up to 10 Gbit/s.

The AX/4000 Generator and Generator/Analyser modules include tools for creating unlimited traffic variations and details. The controller software has a very intuitive graphical user interface, that allow us to build complex traffic streams quickly and easily. When injected onto the network, these traffic streams can be "shaped" (to simulate constant or bursty traffic) and even introduce error conditions. Figure 3.2 shows how the network access graphical user interface (NAGUI) looks like.

For our purpose we will use the control module and the Gigabit Ethernet generator/analyser module that are installed in the AX/4000 chassis of the laboratory. The control module allows the system to be connected to an Ethernet-based LAN for access by remote users.

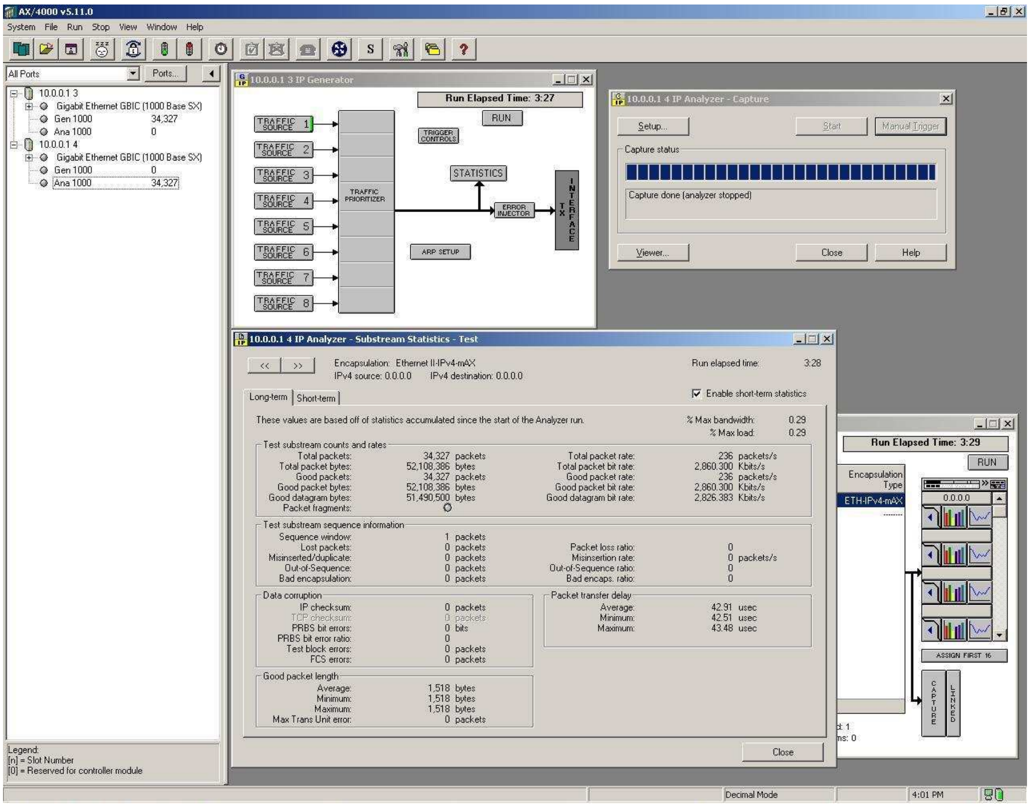

Figure 3.2: Traffic Generator Graphical User Interface

The control module receives the signals coming from the GPS receiver to synchronise the generator with UTC time. It inputs both PPS and 10 MHz signal directly from the Thunderbolt GPS clock through a coaxial cable.

As we explained in section 2.2.4, the PPS signal just indicates the start of a second but not the information about the time. For this reason the control module has an input for serial communication with the GPS receiver. One problem we had to face is that the GPS receivers that we use are not compatible with the GPS serial communication that the traffic generator is expecting. To solve this problem we use a C program to send the time information in the appropriate format. This program is running in the PC containing the card under test, so this PC has its serial port connected to the AX/4000 control module.

The Gigabit Ethernet generator/analyser module uses an optical interface to output and input the traffic. The signal is then converted from optical to electrical in a switch before it gets to the capture card.

3.3.6 Endace DAG Card 3.7G

Endace DAG card has been designed for the capture and generation of Ethernet network packets. Ethernet packets are received through either of the two RJ45 interfaces of the card, and transferred through framers to the FPGA. The FPGA has an Ethernet processor and the synchronisation engine that calculates the timestamps.

Because of close association of the components, packets are time-stamped accurately. Time stamped packet records are stored by the FPGA, which interfaces to the PCI bus. All packet records are written to host PC memory during capture operations.

Figure 3.3 shows the card's major components and the flow data.

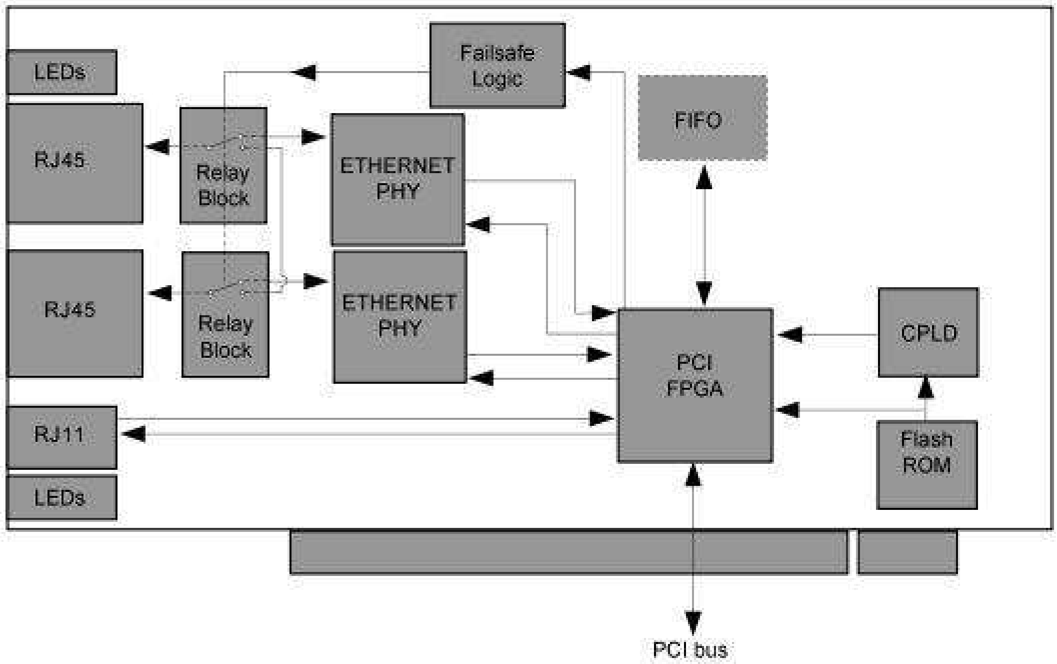

Figure 3.3: DAG card internal structure

DAG Universal Clock System

The Endace DAG cards have sophisticated time synchronisation capabilities, which allow for high quality timestamps, optionally synchronised to an external time standard. The core of the DAG synchronisation capability is known as the DAG Universal Clock Kit (DUCK).

The DUCK is designed to reduce time variance between sets of DAG cards or between DAG cards and coordinated universal time (UTC). You can obtain an accurate time reference by connecting an external clock to the DAG card using the time synchronisation connector. Alternatively you can use the host PCs clock in software as a reference source without any additional hardware. Each DAG card can also output a clock signal for use by other cards.

The DAG card time synchronisation connector supports a Pulse-Per-Second (PPS) input signal. Common synchronisation sources include GPS or CDMA (cellular telephone) time receivers.

DAG Card Behaviour

The main feature of the DAG card is that it has the GPS directly connected to the card. Instead of having the kernel timestamp the arriving/sending packets, timestamping is performed as soon as the packet arrives/leaves by the card itself. As a result, no kernel induced jitter is present in the packet timestamp. Besides, the high-precision card uses a shared memory as a mean to relay packets to the analysis program running in user space, in such a way that interrupts and packet copies are avoided.

Figure 3.4 compares this card with a conventional one in which the timestamping is provided at kernel or user level. It presents the packet transversed through Linux kernel and the points where the packet is timestamped: t1' for transmission (inserted as a new header field in the packet) and t2' for reception. Timestamp t1 is marked when the first bit of the packet is put into the network and t2 is marked when the first bit is received. These both t1 and t2 would be the ideal, however we get different t1' and t2'. For DAG card, the timestamping is done at card level, much near to real t1 and t2. However, for conventional Ethernet card t1' is done by the application in the sending process and t2' by the kernel in the reception process.

Interface

Figure 3.5 shows the measurement setup structure for the DAG card. The DAG card receive the Ethernet packet from the traffic generator in one of the RJ45 ports. Since the traffic generator outputs the packets to a optical link, we need to convert this signal to be carried by an electrical signal. We use an electrical/optical switch for that purpose.

For synchronisation with UTC the card receive a PPS input through a RJ11 connector. It was necessary to build an adaptor from DB9 to RJ11 to connect

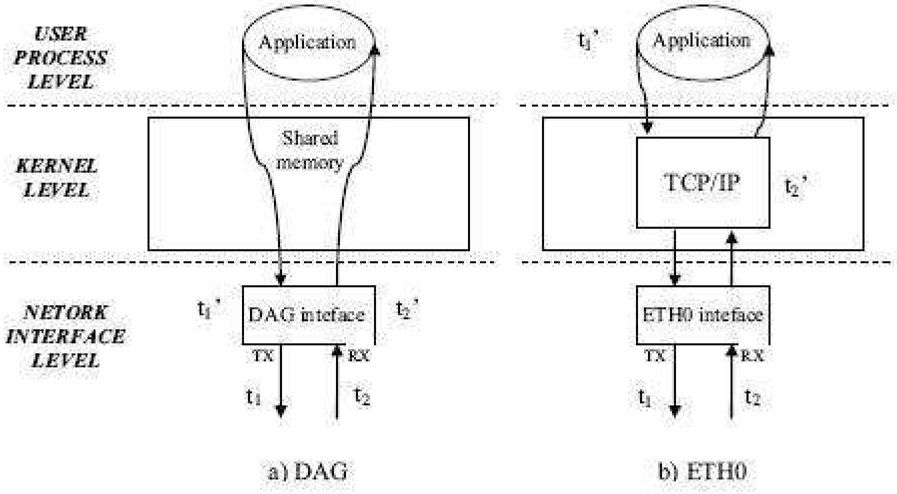

Figure 3.4: Timestamping with DAG card and conventional card.

the distribution board to the DAG card. The pulse has RS-232 signalling levels.

3.3.7 SynPCI Card

As we have seen in section 2.4, there exists a lot of error sources affecting to the accurate synchronisation of the hardware. The biggest problem is the uncertainty in the undetermined delay caused by system interrupts. If we can reduce or eliminate this source of error, we would improve significantly the accuracy of the system.

The change of interrupts priority would require some modification in the hardware so it is not an adequate solution for a wide use. With the PPS signal (section 2.2.4) we obtained an accurate mark of time every second, we can change this to be able to check the actual instant of time when it best suites and when it takes the minimum delay. Based on this, we can perform the appropriate corrections.

The idea is to build an add-on card (SynPCI) that contains two registers. The first register is a counter that is reset when it is read. The other register keeps the value of the first counter when it receives a PPS signal. The source clock for the first counter is a voltage-controlled oscillator (VCO) phase locked with a signal of 10 MHz coming from the Thunderbolt GPS clock.

When a timer interrupt arrives, the 32-bits value of the first register is read.

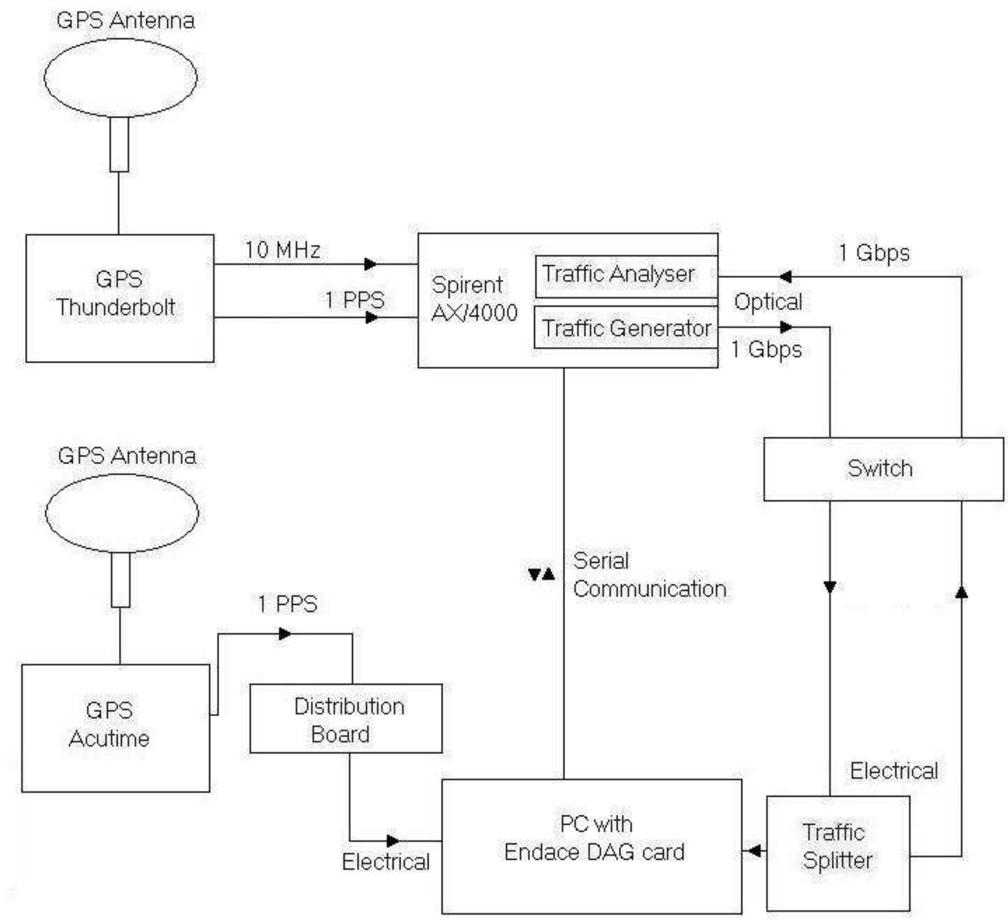

Figure 3.5: Measurement setup for the DAG card.

This value includes 1 bits indicating the arrival of the PPS signal, and a 31-bit value from the first counter. The size of the counter is enough as it is read from 100 to 1000 times per second. The kernel value xtime is updated with the value read from the register.

Now, the PPS indicator is used to detect if a PPS pulse has been received between the timer interrupts. If the bit is activated, then the other register is accessed and a correction to xtime is applied if needed.

Improvements with respect to PPS

Compared with the normal use of the PPS signal, the main feature of this method is that the interrupt latency does not have any influence in the timestamping process. Time between interrupts can not be larger than 2 seconds, but besides that, they can occur in any moment. If the access to the PCI bus is delayed, the precision is not affected because the counter value is recorded only when it is read. The read process lasts about 300 ns in a 133 MHz PCI-X bus and from 450 to 650 ns on 33 MHz PCI bus, depending on the architecture of the motherboard.

Design of the SynPCI card

Figure 3.6 shows the structure of the SynPCI synchronisation system. The design consists of one Lattice LCMXO1200 complex programmable logic device (CPLD). This chip provides PCI bus functions and counter register functions. The card has two optical inputs for the PPS signal and for the 10 MHz signal. The VCO frequency can be set with software to any multiple of 10 MHz, with a maximum of 400 MHz. Selecting a correct multiplication factor for the PLL allows us to have the most suitable solution at 2^N nanoseconds, what results in a faster execution.

The design choice of having optical fibre connections brings us a lower attenuation and a greater transmission distance. Optical fibre features minimum loss of signal power, so that it can transmit data, voice and video at higher speeds and greater distances. Furthermore, optical fibre does not need grounding and prevents from electromagnetic interference.

The SynPCI card has been designed to be installed vertically on 1U rack chassis. The card has LED indicators for the signal inputs in order to verify the correct installation of the optical fibre. For future applications a connector with 20 I/O signals can be used to help with signal distribution or to receive new inputs. This connector and the free capacity of the programmable chip can be used to interconnect it with other systems.

In order to input the optical signals, the SynPCI card also needs a distribution board that converts the electrical PPS signal and the 10 MHz signal from the Thunderbolt GPS clock to the desired number of optical outputs. Figure 3.7 shows the schematic of the measurement setup for the SynPCI card, the interfacing and the data flow for the main components.

The SynPCI card has a cost of 100 euros per card approximately, the distribution board costs about 50 euros per card and the Thunderbolt GPS receptor

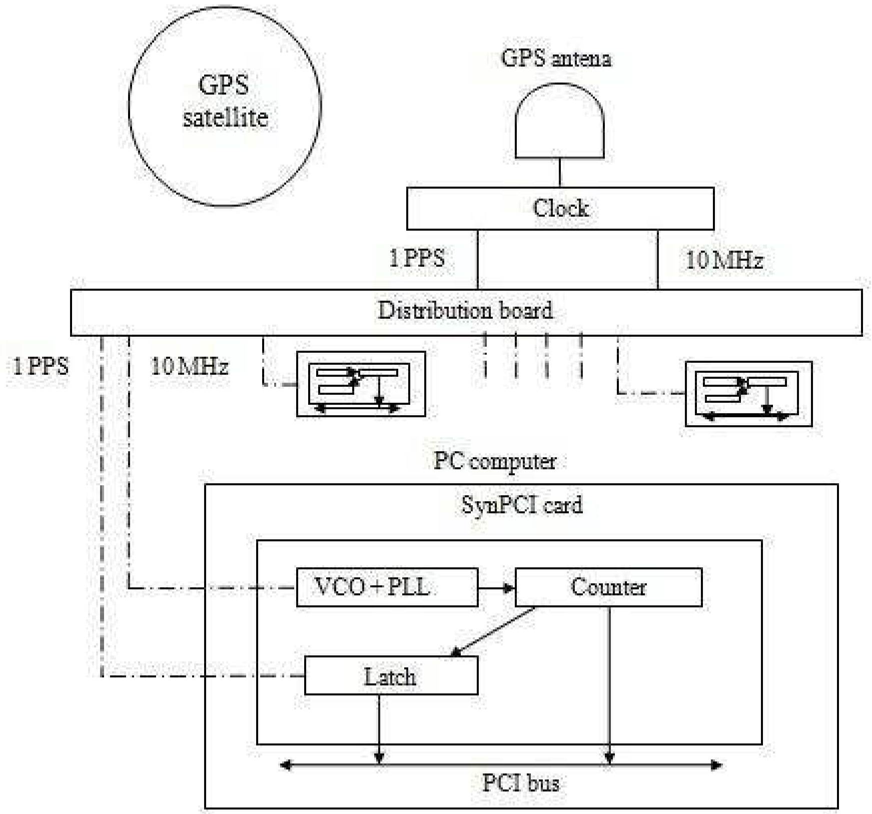

Figure 3.6: Main components of SynPCI synchronisation system

is about 1500 euros. Therefore for a laboratory with 20 PCs synchronised, the total cost of all the equipment is 4500 euros, 225 per PC.

Delay Analysis of SynPCI Architecture

To be able to characterise the performance of the network, we have to study all the components that affect to the system delay. The Thunderbolt GPS clock provides both PPS signal and 10MHz signal. According to manufacturer the accuracy of the PPS signal is 20 ns. It is connected through a 50 ohms coaxial cable to the distribution board. Assuming that the length of the cables is about 2 metres, we can estimate a constant delay of 10 ns.

Converting the electrical signal to optical and back again to electrical also adds up some delay. Experimentally this delay in our implementation has been calculated resulting in 25 ns of delay. The delay added by the optical fibres is about 5 ns per metre.

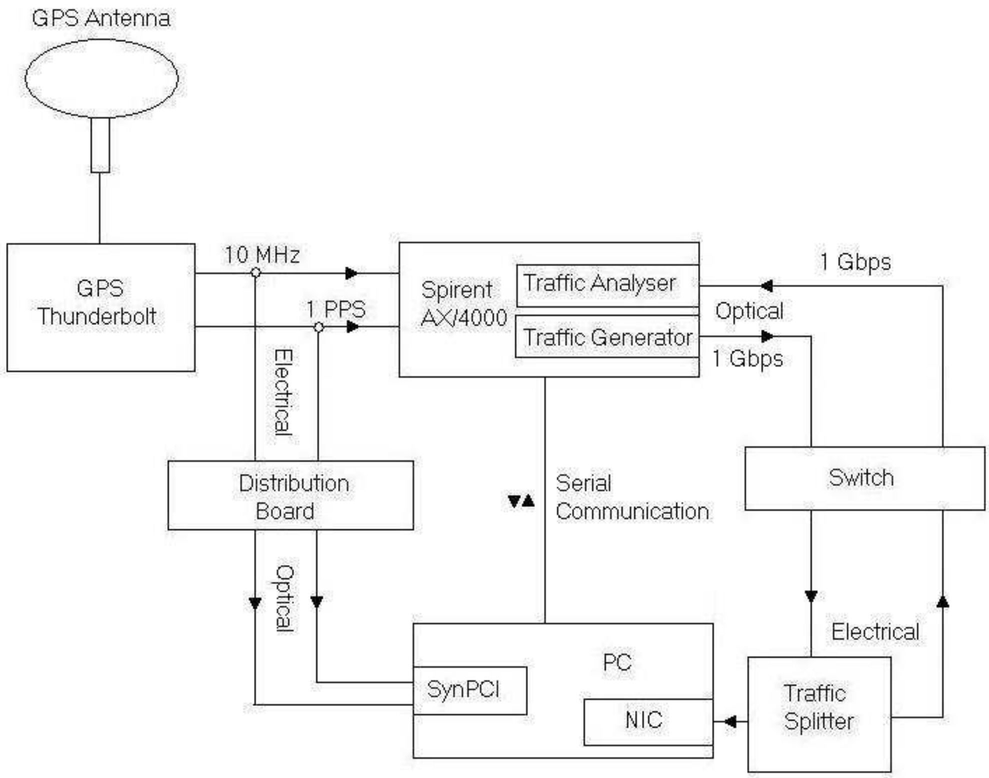

Figure 3.7: Measurement setup for the SynPCI card.

Finally we can conclude that the total delay of the system produced by the signal distribution and conversion is constant and about 85 ns considering 10 metres of optical fibre. [11]

Chapter 4

Analysis

In this chapter we will present the results that have been obtained from different measurements done for both of the devices under test. First we will explain the test procedure, the sets of test that have been performed and their purpose. Then we will show the results themselves and we will analyse the meaning of this results.

4.1 Test Procedure

The test procedure we will follow to analyse the performance of the two capture cards is very simple. As we explained in chapter 3, the traffic signal is sent by the traffic generator using Gigabit Ethernet standard, split before it gets the capture card, and looped back again to the high-performance analyser of the AX/4000 traffic generator. All the packets are timestamped in the card under test and in the AX/4000 analyser.

Our aim is to measure the accuracy provided by the capture cards to monitor the traffic and timestamp the packets sent by the traffic generator. Since it is not possible to measure the accuracy directly, we have to measure the difference between the timestamps given by the capture card and the analyser used as reference.

The AX/4000 network analyser allows us to capture the network packets using the graphic interface. From that interface we can export a text file containing the timestamp information with nanosecond resolution.

For the DAG card we use the command *dagsnap* to start capturing the traffic. It returns a file in erf format readable by some network protocol analyser programs like Wireshark. The resolution of the timestamps is 1 ns.

The PC containing the synPCI card uses the TCPdump tool to perform the capture. Unfortunately it only provides microsecond accuracy, decreasing the quality of the results compared with the results given by the DAG card.

4.1.1 Network Conditions

We will operate in a very small network where all the links are controlled and the only existing traffic is the one we generate. This allows us to know in every moment the exact condition of the network. We will also suppose that there are not errors in the transmission and all the packets sent are able to reach their destination.

4.1.2 Test Cases

The traffic pattern we have used for the tests consists in a burst of periodic packets with fixed length. We are interested in knowing how the accuracy of the timestamps is affected with the density of traffic and with the size of the packets. Therefore, we have performed different tests with different packet sizes and different bit rates.

Three different traffic sizes and three different bit rates have been combined to perform the tests, therefore nine different traffic patterns have been used to test each capture card. The packet sizes that have been chosen are 46, 500 and 1400 bytes of IP protocol, adding 18 bytes corresponding to Ethernet header and trailer. This choice have been taken because we can see the effects of using the minimum packet size, a medium one and another near to the Maximum Transfer Unit (MTU) of the Ethernet protocol.

The bit rates used have been selected to be able to study the behaviour of the capture cards with a low bit rate and with higher traffic speeds. For this reason we have run the tests using bit rates of 10 Mbit/s, 200 Mbit/s and 500 Mbit/s, what means a traffic density of 1%, 20%, and 50% of the link capacity respectively.

To perform the captures, the AX/4000 network analyser has a buffer of 64 MB. This set a limit on the duration of the captures that we want to do. For that reason captures done at a low transmission rate will be longer than captures done with a high transmission rate.

4.2 Test Results

First we will present the results for the DAG card and the synPCI card for the different transmission bit rates and packet lengths. Then we will see other results and considerations.

4.2.1 DAG Card Results

For each case the mean, 1^{st}-99^{th} interpercentile range and round trip time, wich is calculated directly in the AX/4000 analyser, are presented. Moreover two different graphs are included. One of them shows the evolution of the delay during the capture. The other is a histogram showing the distribution of the delay.

DAG card: 64 bytes packets at 10 Mbit/s

The traffic generator sends a traffic burst consisting in 720874 periodic packets with a fixed length of 64 bytes (46 bytes IP packet + 18 bytes Ethernet header and trailer). The transmission bit rate is 10 Mbit/s (1% of the link capacity), the duration of the burst is 36.91 seconds and the interarrival time is 55.36 μs.

Table 4.1 shows the round trip time, mean delay and other statistical information. We can observe that even if the mean delay value is around 3.3 microseconds, the interpercentile range between the 1^{st} and the 99^{th} percentile is only 185 nanoseconds. We use this interpercentile range and not the whole range to get rid of extreme values and to avoid outliers.

Table 4.1: DAG card statistics for 64 bytes packets at 10 Mbit/s

Minimum delay	3.141 μs
Maximum delay	3.430 μs
Mean	3.303 μs
99^{th} percentile	3.384 μs
1^{st}-99^{th} interpercentile range	185 ns
Standard deviation	39.45 ns
Round Trip Time	7.24 μs

The value of the mean is caused by the delay that suffer the packet in order to reach the DAG card from the traffic generator. We can check that this mean value is approximately the half of the round trip time given at the traffic generator.

The distribution of the delay is represented in the histogram shown in figure 4.1. Most of the observations are concentrated around the mean value, with two short tails in both sides.

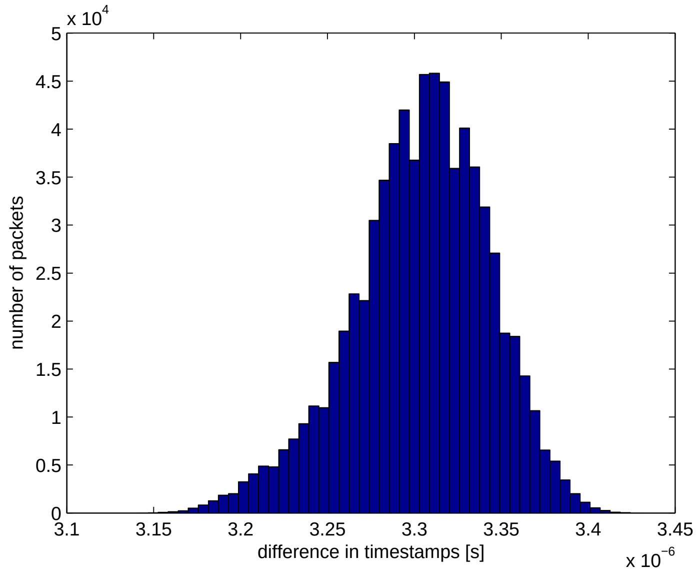

Figure 4.1: Histogram for 64 bytes packets at 10 Mbit/s captured by the DAG card

Figure 4.2 shows how the delay behaves during the time of the capture. The capture has been divided in 100 intervals. The figure shows the maximum, minimum and average delay for each of these intervals. In this way we are able to see the evolution of the delay regarding to the average and watch if there exists any outlier observation.

We can see that the delay changes more or less abruptly in some points during the capture. This is due to the corrections made in the capture card when it receives the PPS signal. In a more detailed view, shown in figure 4.3, we can appreciate that the changes in the slope happens precisely in every change of second.

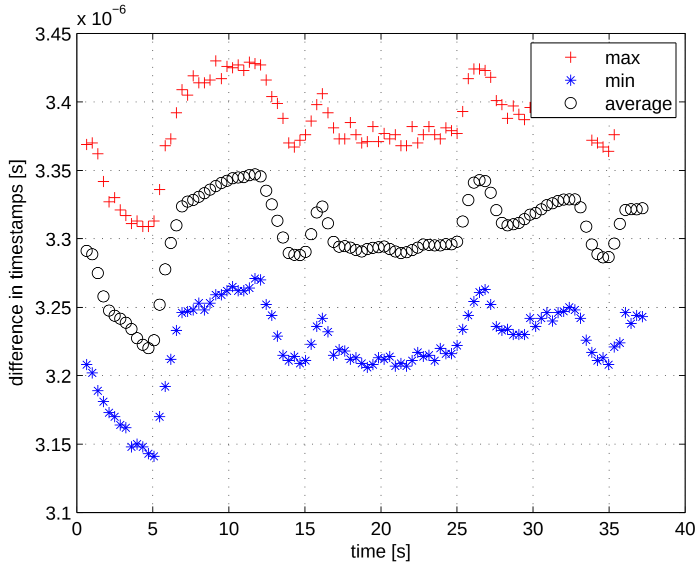

Figure 4.2: Delay evolution for 64 bytes packets at 10 Mbit/s captured by the DAG card

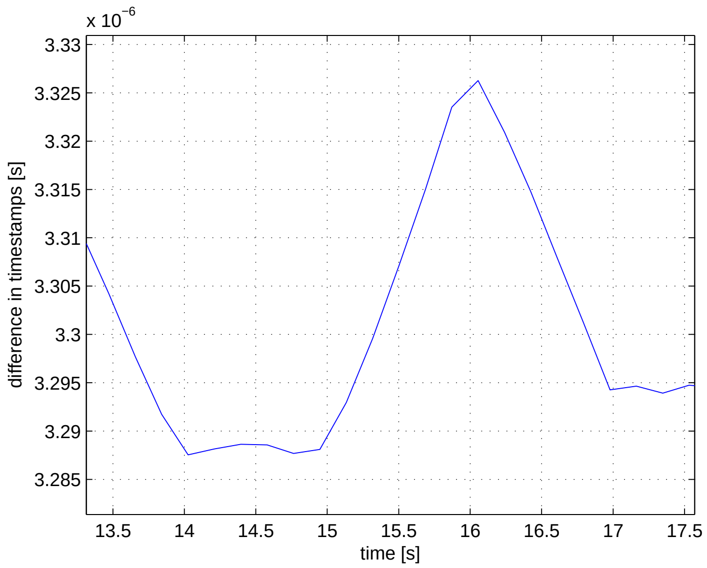

Figure 4.3: Detail of the changes in the delay

DAG card: 64 bytes packets at 200 Mbit/s

The traffic generator sends a traffic burst consisting in 720874 periodic packets with a fixed length of 64 bytes (46 bytes IP packet + 18 bytes Ethernet header and trailer). The transmission bit rate is 200 Mbit/s (20% of the link capacity), the duration of the burst is 1.845 seconds and the packet interarrival time is 2.56 μs.

Table 4.2 shows the round trip time, mean delay and other statistical information. All the data, except for the dispersion values, is approximately the same than the previous case. The interpercentile range is substantially lower in this case, we can deduce that this happens because the duration of the experiment is 20 times shorter and therefore less corrections in the local time have been done, what results in a lower dispersion on the observation set.

Table 4.2: DAG card statistics for 64 bytes packets at 200 Mbit/s

Minimum delay	3.161 μs
Maximum delay	3.405 μs
Mean	3.291 μs
99^{th} percentile	3.356 μs
1^{st}-99^{th} interpercentile range	129 ns
Standard deviation	28.39 ns
Round Trip Time	7.24 μs

The distribution of the delay is represented in the histogram shown in figure 4.4. As we said, in this case the dispersion of the data is lower and therefore the observations are more concentrated and the tails are smaller.

Figure 4.5 shows how the delay behaves during the time of the capture. The capture has been divided in 100 intervals. The figure shows the maximum, minimum and average delay for each of these intervals.

DAG card: 64 bytes packets at 500 Mbit/s

The traffic generator sends a traffic burst consisting in 720874 periodic packets with a fixed length of 64 bytes (46 bytes IP packet + 18 bytes Ethernet header and trailer). The transmission bit rate is 500 Mbit/s (50% of the link capacity), the duration of the burst is 0.738 seconds and the packet interarrival time is 1.02 μs.

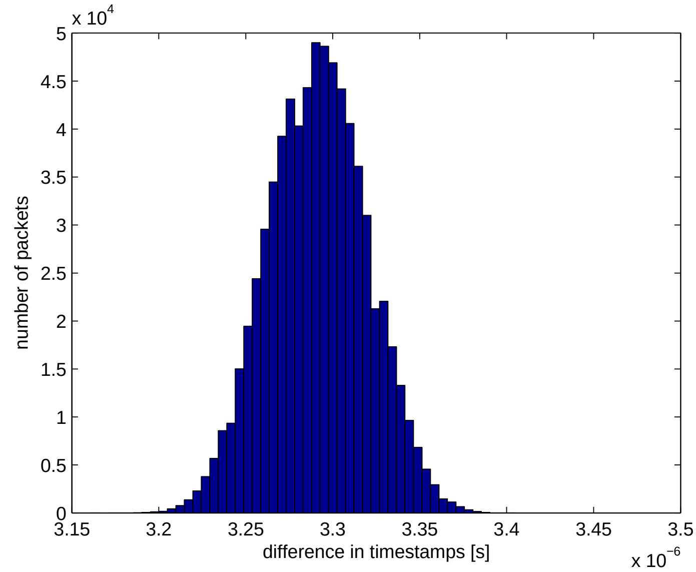

Figure 4.4: Histogram for 64 bytes packets at 200 Mbit/s captured by the DAG card

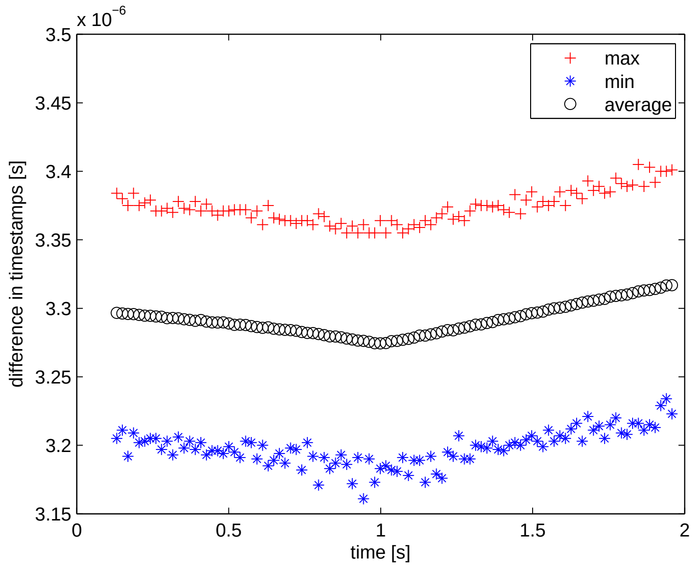

Figure 4.5: Delay evolution for 64 bytes packets at 200 Mbit/s captured by the DAG card

Table 4.3 shows the statistical information that is very similar than the one in the previous case. We observe that even with a high transmission rate, the behaviour of the DAG card does not change significantly.

Table 4.3: DAG card statistics for 64 bytes packets at 500 Mbit/s

Minimum delay	3.196 μs
Maximum delay	3.41 μs
Mean	3.309 μs
99^{th} percentile	3.37 μs
1^{st}-99^{th} interpercentile range	127 ns
Standard deviation	27.89 ns
Round Trip Time	7.24 μs

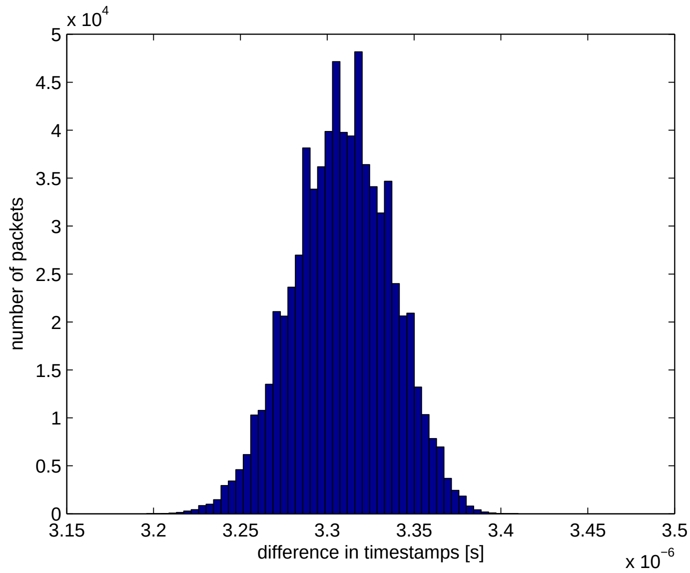

Figure 4.6: Histogram for 64 bytes packets at 500 Mbit/s captured by the DAG card

The distribution of the delay is represented in the histogram shown in figure 4.6 and figure 4.7 shows the evolution of the delay during the time of the capture.

We can appreciate that a PPS pulse is received during the capture since there is one change in the slope of the average evolution.

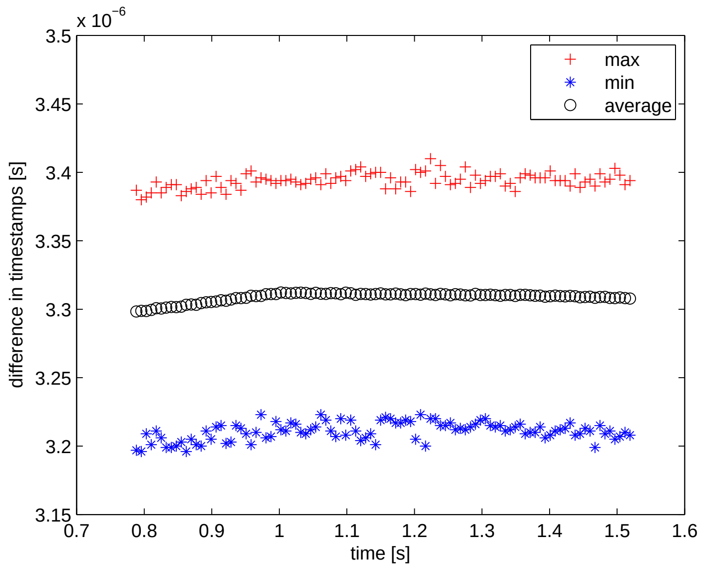

Figure 4.7: Delay evolution for 64 bytes packets at 500 Mbit/s captured by the DAG card

DAG card: 518 bytes packets at 10 Mbit/s

The traffic generator sends a traffic burst consisting in 98300 periodic packets with a fixed length of 518 bytes (500 bytes IP packet + 18 bytes Ethernet header and trailer). The transmission bit rate is 10 Mbit/s (1% of the link capacity), the duration of the burst is 36.91 seconds and the packet interarrival time is 375.5 μs.

Table 4.4 shows the round trip time, mean delay and other statistical information. The mean delay is bigger than for the 64 bytes packets but it is still caused by the fix transmission delay of the link, and it is just a little bit smaller than the half of the round trip time.

Table 4.4: DAG card statistics for 518 bytes packets at 10 Mbit/s

Minimum delay	8.603 μs
Maximum delay	8.858 μs
Mean	8.731 μs
99^{th} percentile	8.803 μs
1^{st}-99^{th} interpercentile range	148 ns
Standard deviation	32.28 ns
Round Trip Time	18.26 μs

The distribution of the delay is represented in the histogram shown in figure 4.8, and the delay evolution is shown in figure 4.9.

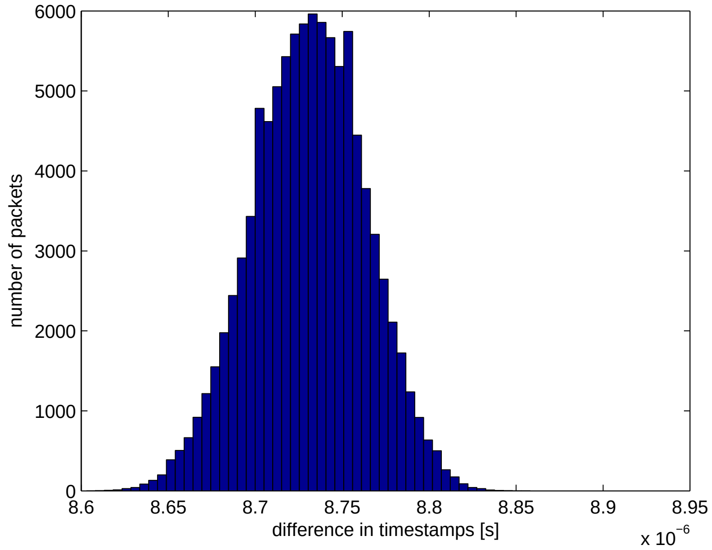

Figure 4.8: Histogram for 518 bytes packets at 10 Mbit/s captured by the DAG card

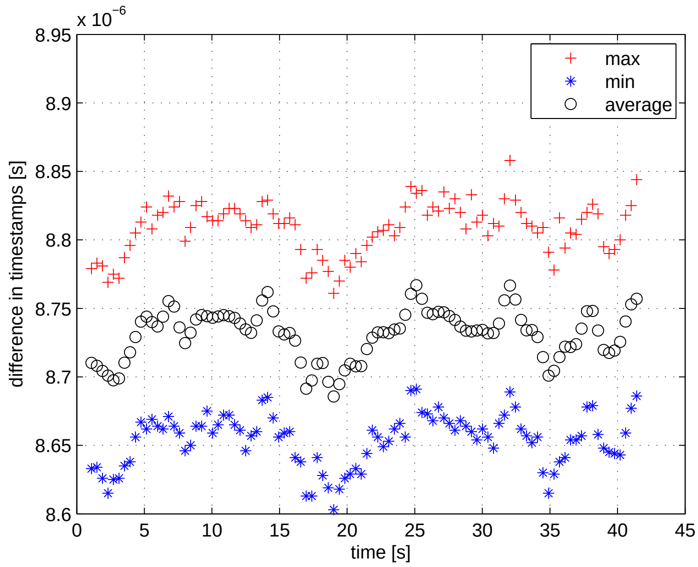

Figure 4.9: Delay evolution for 518 bytes packets at 10 Mbit/s captured by the DAG card

DAG card: 518 bytes packets at 200 Mbit/s

The traffic generator sends a traffic burst consisting in 98300 periodic packets with a fixed length of 518 bytes (500 bytes IP packet + 18 bytes Ethernet header and trailer). The transmission bit rate is 200 Mbit/s (20% of the link capacity), the duration of the burst is 1.845 seconds and the packet interarrival time is 18.77 μs.

Table 4.5 shows the round trip time, mean delay and other statistical information. We can observe that the minimum maximum and mean delay are approximately the same than the previous case. The interpercentile range is lower due to the smaller duration of the observation.

Table 4.5: DAG card statistics for 518 bytes packets at 200 Mbit/s

Minimum delay	8.657 μs
Maximum delay	8.85 μs
Mean	8.753 μs
99^{th} percentile	8.813 μs
1^{st}-99^{th} interpercentile range	120 ns
Standard deviation	26.91 ns
Round Trip Time	18.26 μs

The distribution of the delay and the evolution of the maximum minimum and mean delay are shown in figures 4.10 and 4.11.

DAG card: 518 bytes packets at 500 Mbit/s captured by the DAG card

The traffic generator sends a traffic burst consisting in 98300 periodic packets with a fixed length of 518 bytes (500 bytes IP packet + 18 bytes Ethernet header and trailer). The transmission bit rate is 500 Mbit/s (50% of the link capacity), the duration of the burst is 0.738 seconds and the packet interarrival time is 7.508 μs.

Table 4.6 shows the round trip time, mean delay and other statistical information. Again the higher transmission rate does not affect to the performance of the DAG card, as we can see the statistical values extracted from this observation are very similar than in the previous one.

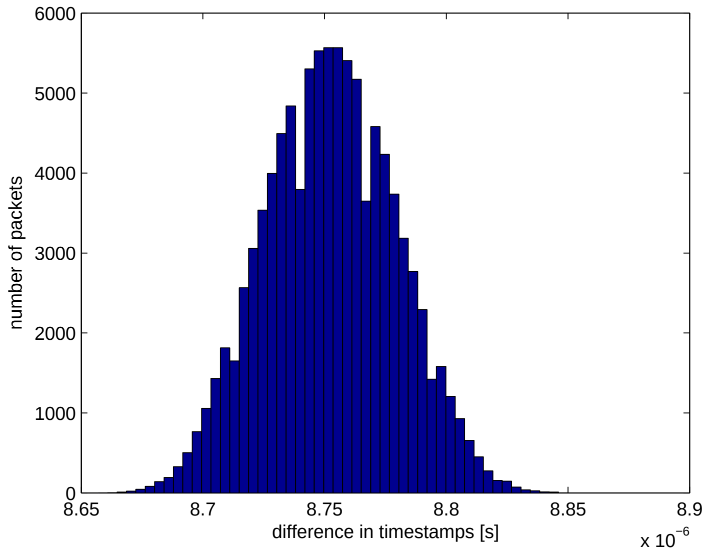

Figure 4.10: Histogram for 518 bytes packets at 200 Mbit/s

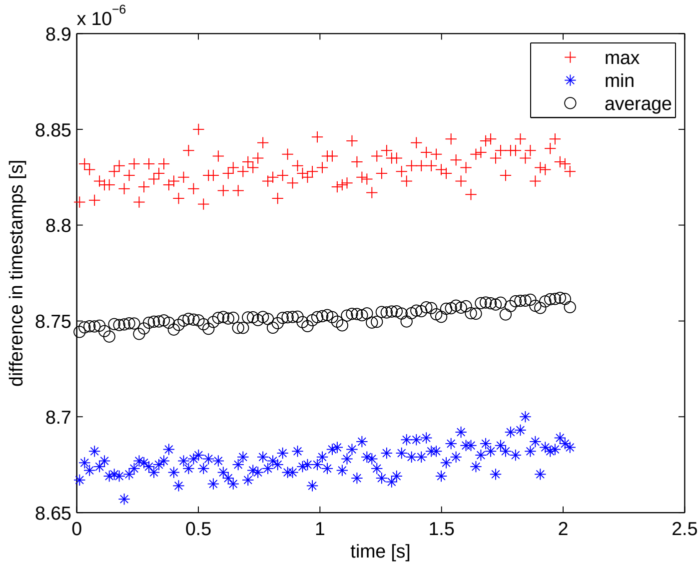

Figure 4.11: Delay evolution for 518 bytes packets at 200 Mbit/s captured by the DAG card

Table 4.6: DAG card statistics for 518 bytes packets at 500 Mbit/s

Minimum delay	8.665 μs
Maximum delay	8.856 μs
Mean	8.755 μs
99^{th} percentile	8.816 μs
1^{st}-99^{th} interpercentile range	121 ns
Standard deviation	26.77 ns
Round Trip Time	18.26 μs

The distribution of the delay and the evolution of the maximum minimum and mean delay are shown in figures 4.12 and 4.13.

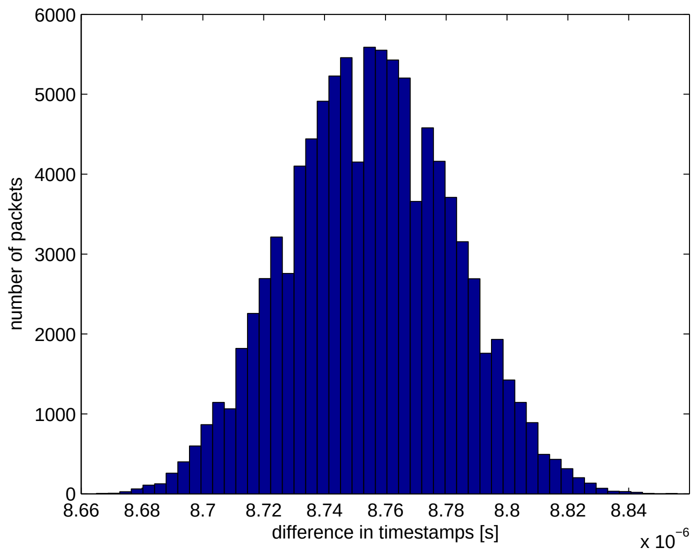

Figure 4.12: Histogram for 518 bytes packets at 500 Mbit/s captured by the DAG card

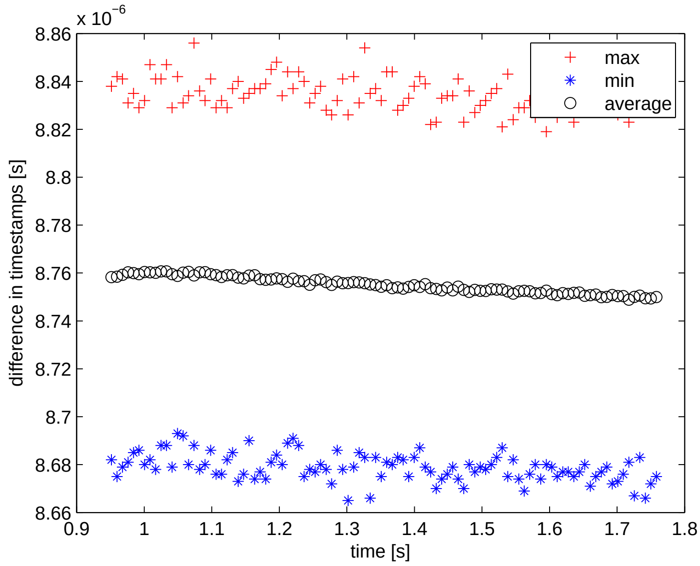

Figure 4.13: Delay evolution for 518 bytes packets at 500 Mbit/s captured by the DAG card

DAG card: 1418 bytes packets at 10 Mbit/s

The traffic generator sends a traffic burst consisting in 36000 periodic packets with a fixed length of 1418 bytes (1400 bytes IP packet + 18 bytes Ethernet header and trailer). The transmission bit rate is 10 Mbit/s (1% of the link capacity), the duration of the burst is 36.91 seconds and the packet interarrival time is 1.025 ms.

Table 4.7 shows the round trip time, mean delay and other statistical information. The round trip time has raised since the size of the packets sent is bigger, and therefore the minimum maximum and mean delay has increased their value in the same way.

Table 4.7: DAG card statistics for 1418 bytes packets at 10 Mbit/s

Minimum delay	20.035 μs
Maximum delay	20.273 μs
Mean	20.160 μs
99^{th} percentile	20.228 μs
1^{st}-99^{th} interpercentile range	139 ns
Standard deviation	30.61 ns
Round Trip Time	40.61 μs

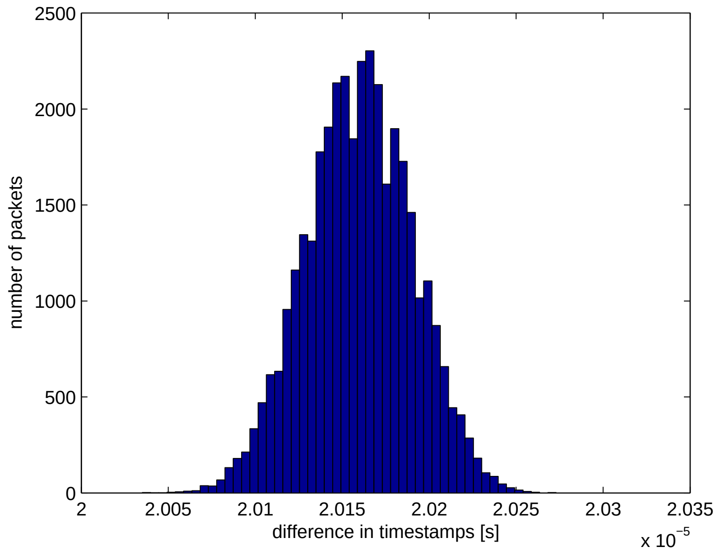

Figure 4.14: Histogram for 1418 bytes packets at 10 Mbit/s captured by the DAG card

The distribution of the delay and the evolution of the maximum minimum and mean delay are shown in figures 4.14 and 4.15.

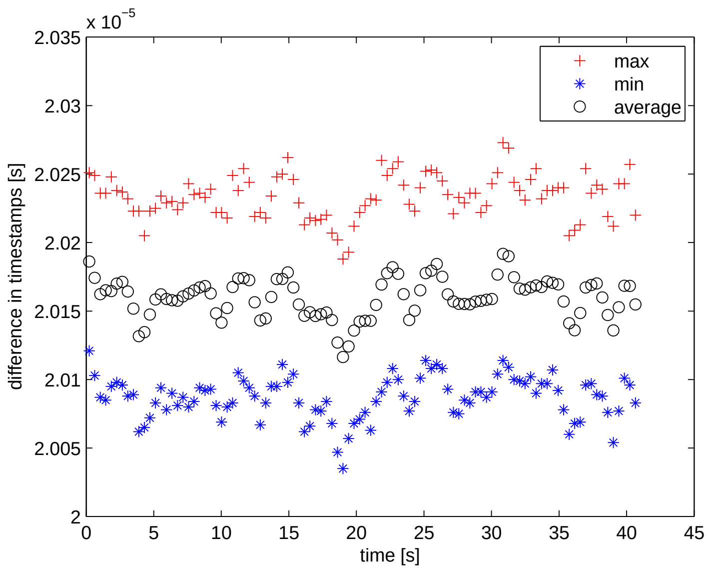

Figure 4.15: Delay evolution for 1418 bytes packets at 10 Mbit/s captured by the DAG card

DAG card: 1418 bytes packets at 200 Mbit/s

The traffic generator sends a traffic burst consisting in 36000 periodic packets with a fixed length of 1418 bytes (1400 bytes IP packet + 18 bytes Ethernet header and trailer). The transmission bit rate is 200 Mbit/s (20% of the link capacity), the duration of the burst is 1.845 seconds and the packet interarrival time is 51.25 μs.

Table 4.8: DAG card statistics for 1418 bytes packets at 200 Mbit/s

Minimum delay	19.999 μs
Maximum delay	20.203 μs
Mean	20.108 μs
99^{th} percentile	20.17 μs
1^{st}-99^{th} interpercentile range	125 ns
Standard deviation	27.81 ns
Round Trip Time	40.61 μs

Table 4.8 shows the round trip time, mean delay and other statistical information. We can see again that with a shorter observation time we have a smaller dispersion because less PPS pulses are received during the capture and less time corrections are performed. The distribution of the delay and the evolution of the maximum minimum and mean delay are shown in figures 4.16 and 4.17.

DAG card: 1418 bytes packets at 500 Mbit/s

The traffic generator sends a traffic burst consisting in 36000 periodic packets with a fixed length of 1418 bytes (1400 bytes IP packet + 18 bytes Ethernet header and trailer). The transmission bit rate is 500 Mbit/s (50% of the link capacity), the duration of the burst is 0.738 seconds and the packet interarrival time is 20.50 μs.

Table 4.9 shows the round trip time, mean delay and other statistical information.

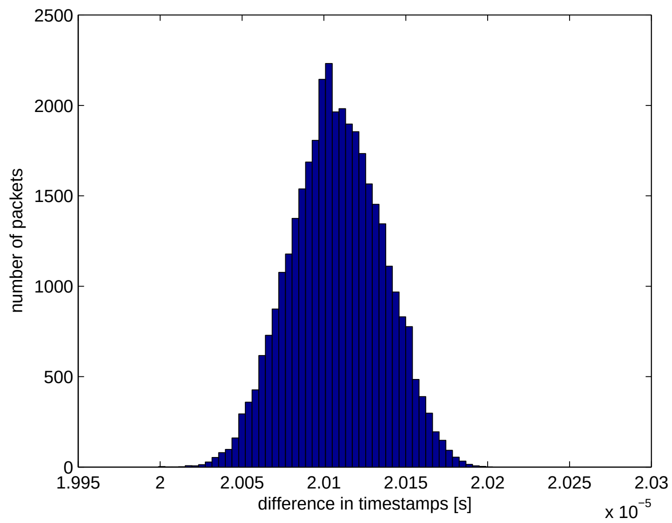

Figure 4.16: Histogram for 1418 bytes packets at 200 Mbit/s captured by the DAG card

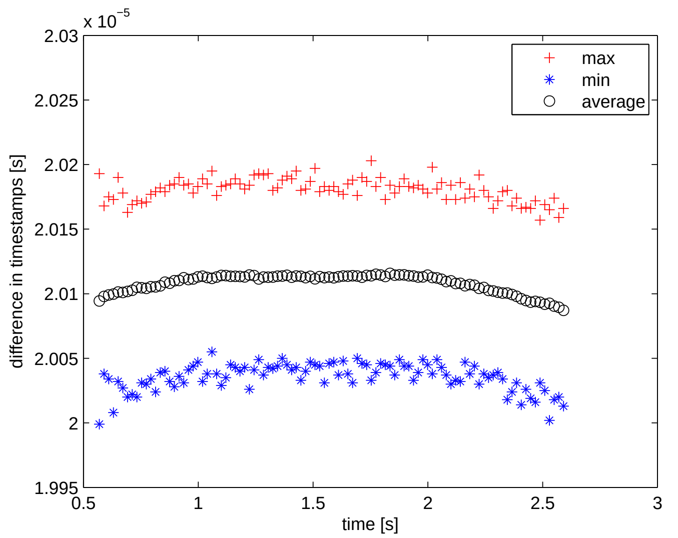

Figure 4.17: Delay evolution for 1418 bytes packets at 200 Mbit/s captured by the DAG card

Table 4.9: DAG card statistics for 1418 bytes packets at 500 Mbit/s

Minimum delay	19.813 μs
Maximum delay	19.999 μs
Mean	19.90 μs
99^{th} percentile	19.97 μs
1^{st}-99^{th} interpercentile range	122 ns
Standard deviation	27.18 ns
Round Trip Time	40.61 μs

The distribution of the delay and the evolution of the maximum minimum and mean delay are shown in figures 4.18 and 4.19.

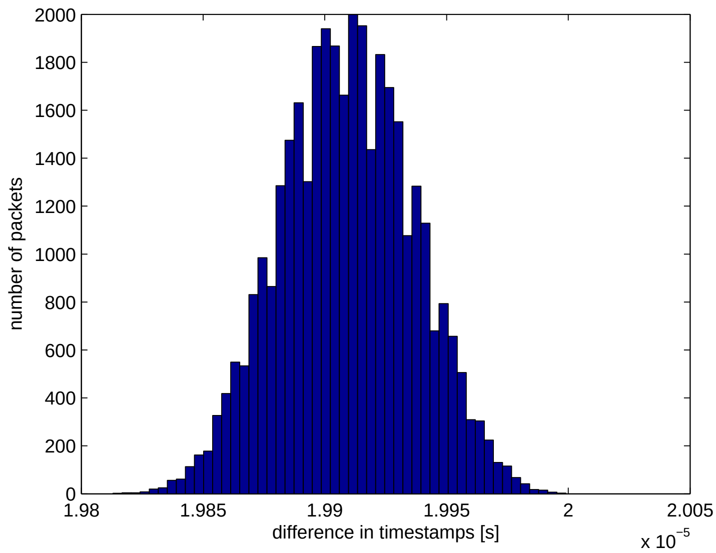

Figure 4.18: Histogram for 1418 bytes packets at 500 Mbit/s captured by the DAG card

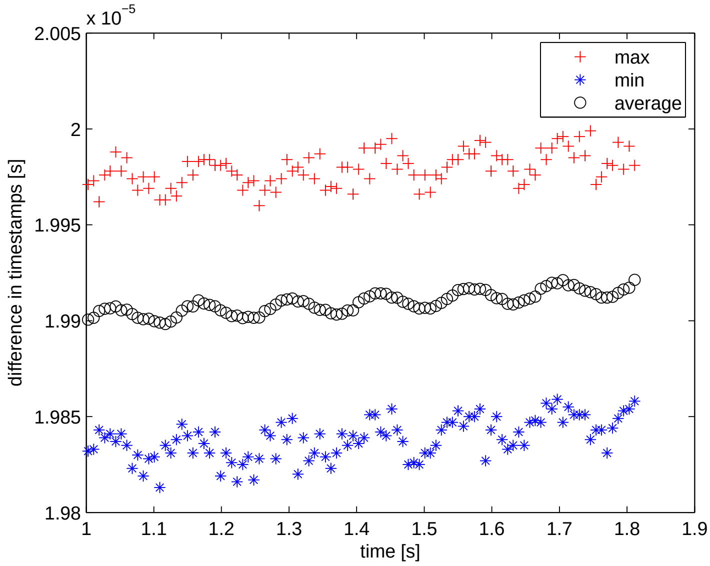

Figure 4.19: Delay evolution for 1418 bytes packets at 500 Mbit/s captured by the DAG card

We can see a little oscillation superposed on the delay evolution probably due to interferences caused by other equipments on the time delay.

4.2.2 SynPCI results

We will present the results obtained for the SynPCI card in the same format than for the DAG card. However, the quality of the results has decreased since the TCPdump tool used to capture the packets only returns timestamps with microseconds resolution.

SynPCI: 64 bytes packets at 10 Mbit/s

The traffic generator sends a traffic burst consisting in 720874 periodic packets with a fixed length of 64 bytes (46 bytes IP packet + 18 bytes Ethernet header and trailer). The transmission bit rate is 10 Mbit/s (1% of the link capacity), the duration of the burst is 36.91 seconds and the interarrival time is 55.36 μs

Table 4.10 shows the round trip time, mean delay and other statistical information. Both mean delay and dispersion values are very much higher than the results taken for the DAG card, what indicates that the performance of the synPCI card is not as good as the DAG card. The delay cannot be explained now just with the round trip time of the signal but there exists other delay caused by the processing time used by the computer containing the synPCI card to timestamp the packets.

We can also see that the maximum delay is a very high value but looking at the 1^{st}-99^{th} interpercentile range we see that it is probably due to some outliers. The round trip time is obviously the same as is not affected by the capture card.

Table 4.10: SynPCI card statistics for 64 bytes packets at 10 Mbit/s

Minimum delay	13.6 μs
Maximum delay	122.3 μs
Mean	14.92 μs
99^{th} percentile	15.97 μs
1^{st}-99^{th} interpercentile range	2.08 μs
Standard deviation	736 ns
Round Trip Time	7.24 μs

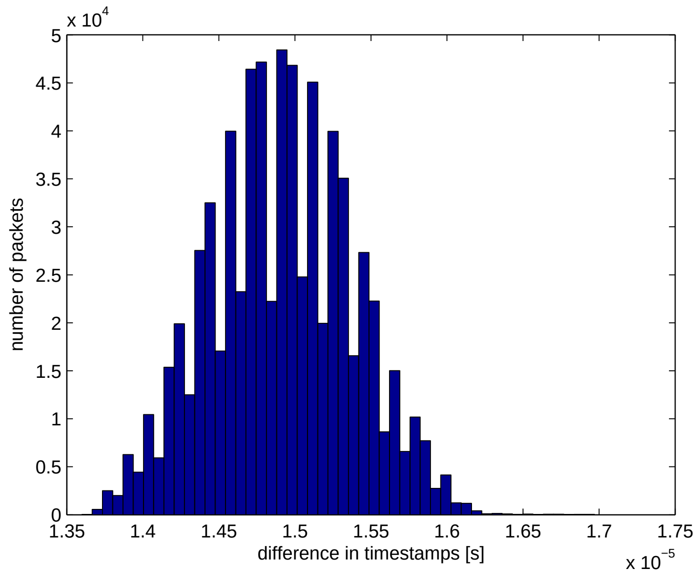

Figure 4.20: Histogram for 64 bytes packets at 10 Mbit/s captured by the synPCI card

The distribution of the delay (without outliers) and the evolution of the maximum minimum and mean delay are shown in figures 4.20 and 4.21.

Here we can see that almost all the timestamps are concentrated around the mean with a little dispersion but there are also some outliers situated far from the mean.

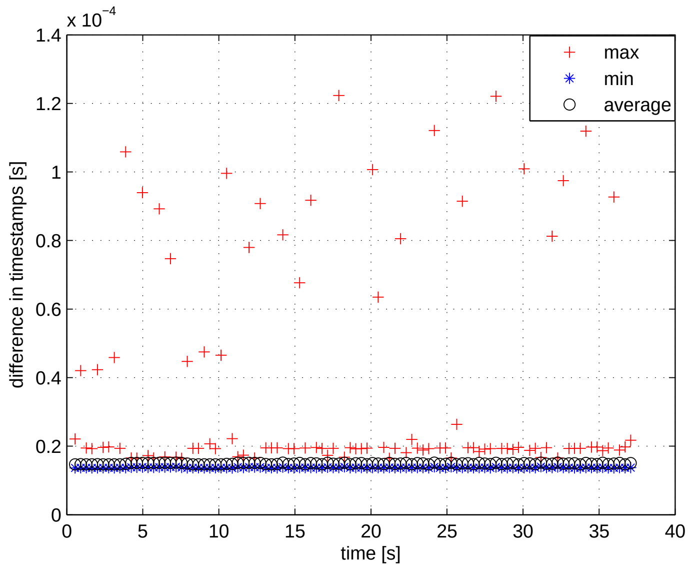

Figure 4.21: Delay evolution for 64 bytes packets at 10 Mbit/s captured by the synPCI card

SynPCI: 64 bytes packets at 200 Mbit/s

The traffic generator sends a traffic burst consisting in 720874 periodic packets with a fixed length of 64 bytes (46 bytes IP packet + 18 bytes Ethernet header and trailer). The transmission bit rate is 200 Mbit/s (20% of the link capacity), the duration of the burst is 1.845 seconds and the packet interarrival time is 2.56 μs.

Table 4.11: SynPCI card statistics for 64 bytes packets at 200 Mbit/s

Minimum delay	7.55 μs
Maximum delay	122.35 μs
Mean	12.265 μs
99^{th} percentile	15.85 μs
1^{st}-99^{th} interpercentile range	7.25 μs
Standard deviation	2.328 μs
Round Trip Time	7.24 μs

Table 4.11 shows the round trip time, mean delay and other statistical information. The maximum delay, the mean and the 99^{th} percentile are approximately the same but the minimum delay is lower than the previous case. This cause the dispersion parameters to increase their values.

The distribution of the delay (without outliers) and the evolution of the maximum minimum and mean delay are shown in figures 4.22 and 4.23. We can see than the distribution is wider due to the increase of the dispersion.

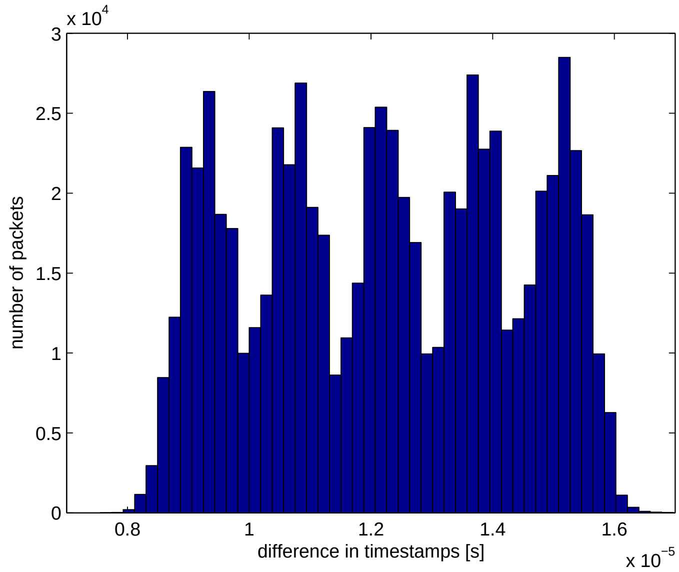

Figure 4.22: Histogram for 64 bytes packets at 200 Mbit/s captured by the synPCI card

SynPCI: 64 bytes packets at 500 Mbit/s

For this case, 64 bytes packets sent with a data transmission rate of 500 Mbit/s, the number of packets received per second was too high and the synPCI card was not able to capture and timestamp all the packets. For that reason we do not have reliable data for this case and we cannot perform the analysis due to this limitation of the capture card.

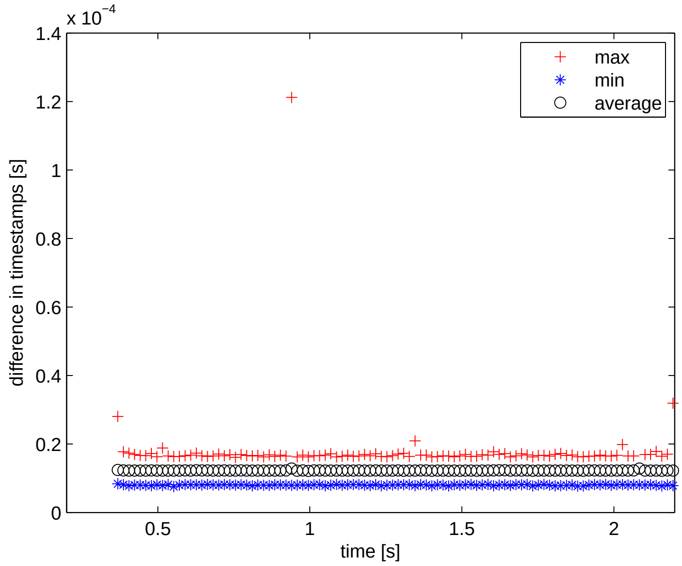

Figure 4.23: Delay evolution for 64 bytes packets at 200 Mbit/s captured by the synPCI card

SynPCI: 518 bytes packets at 10 Mbit/s

The traffic generator sends a traffic burst consisting in 98300 periodic packets with a fixed length of 518 bytes (500 bytes IP packet + 18 bytes Ethernet header and trailer). The transmission bit rate is 10 Mbit/s (1% of the link capacity), the duration of the burst is 36.91 seconds and the packet interarrival time is 375.5 μs.

Table 4.12 shows the round trip time, mean delay and other statistical information. The mean delay is bigger than for the 64 bytes packets, in part because the bigger size of the packets produce a higher transmission delay but the dispersion parameters are just a little higher than before.

Table 4.12: SynPCI card statistics for 518 bytes packets at 10 Mbit/s

Minimum delay	23.05 μs
Maximum delay	116.3 μs
Mean	24.346 μs
99^{th} percentile	25.47 μs
1^{st}-99^{th} interpercentile range	2.17 μs
Standard deviation	776.35 ns
Round Trip Time	18.26 μs

The distribution of the delay (without outliers) and the evolution of the maximum minimum and mean delay are shown in figures 4.24 and 4.25.

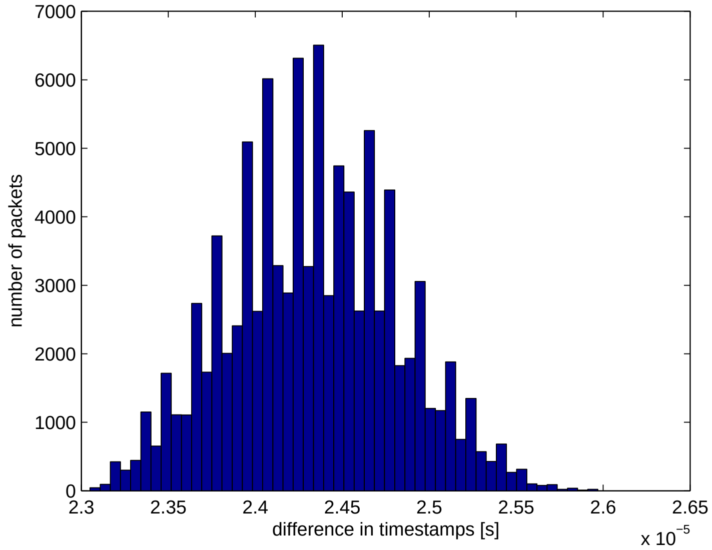

Figure 4.24: Histogram for 518 bytes packets at 10 Mbit/s captured by the synPCI card

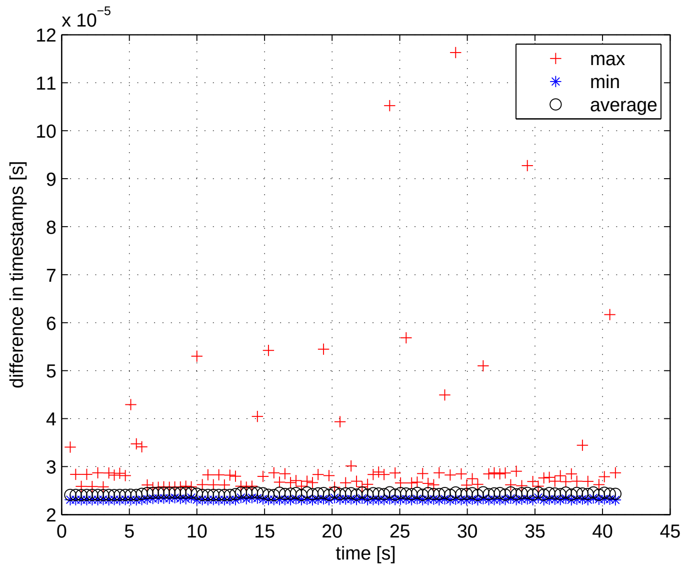

Figure 4.25: Delay evolution for 518 bytes packets at 10 Mbit/s captured by the synPCI card

SynPCI: 518 bytes packets at 200 Mbit/s

The traffic generator sends a traffic burst consisting in 98300 periodic packets with a fixed length of 518 bytes (500 bytes IP packet + 18 bytes Ethernet header and trailer). The transmission bit rate is 200 Mbit/s (20% of the link capacity), the duration of the burst is 1.845 seconds and the packet interarrival time is 18.77 μs.

Table 4.13 shows the round trip time, mean delay and other statistical information. The minimum, maximum and mean delay are higher for this transmission rate, and although the interpercentile range is lower, the standard deviation is higher than for the 10 Mbit/s case.

Table 4.13: SynPCI card statistics for 518 bytes packets at 200 Mbit/s

Minimum delay	24.14 μs
Maximum delay	134.97 μs
Mean	25.547 μs
99^{th} percentile	26.39 μs
1^{st}-99^{th} interpercentile range	1.54 μs
Standard deviation	823.2 ns
Round Trip Time	18.26 μs

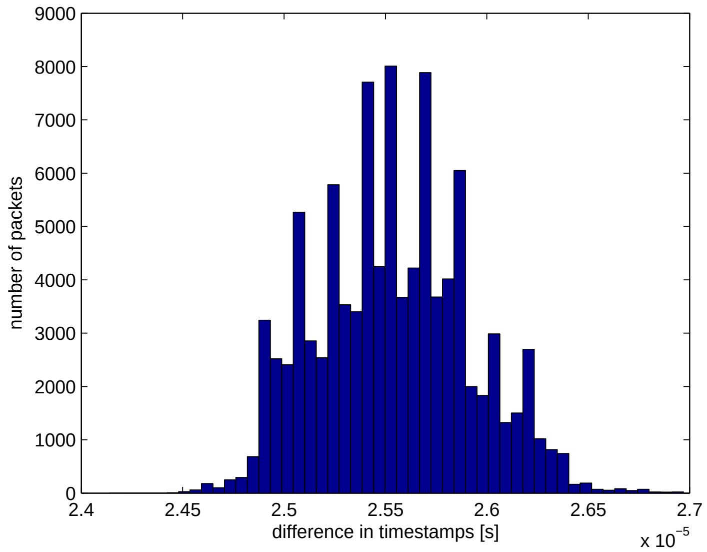

Figure 4.26: Histogram for 518 bytes packets at 200 Mbit/s captured by the synPCI card

The distribution of the delay (without outliers) and the evolution of the maximum minimum and mean delay are shown in figures 4.26 and 4.27. We can see the presence of some outliers.

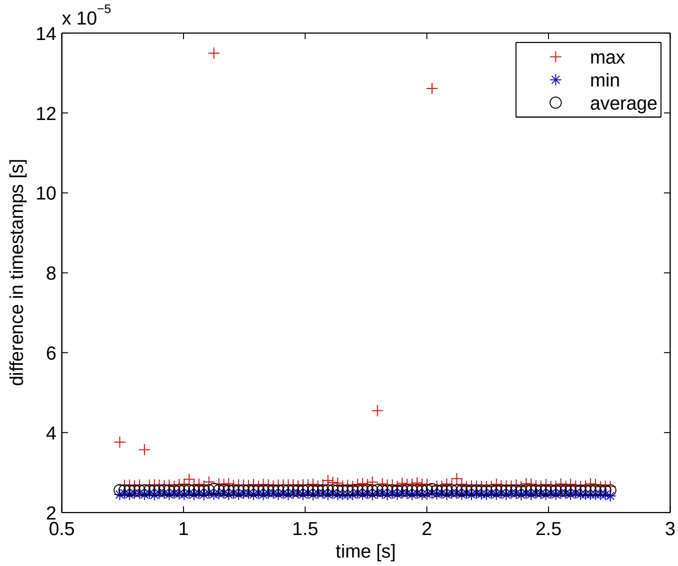

Figure 4.27: Delay evolution for 518 bytes packets at 200 Mbit/s captured by the synPCI card

SynPCI: 518 bytes packets at 500 Mbit/s

The traffic generator sends a traffic burst consisting in 98300 periodic packets with a fixed length of 518 bytes (500 bytes IP packet + 18 bytes Ethernet header and trailer). The transmission bit rate is 500 Mbit/s (50% of the link capacity), the duration of the burst is 0.738 seconds and the packet interarrival time is 7.508 μs.

Table 4.14 shows the round trip time, mean delay and other statistical information. In this case we can see more obviously the influence of the higher transmission rate on the synPCI card. Although the mean is approximately the same, both the standard deviation and the 1^{st}-99^{th} interpercentile range are very much higher than the previous cases.

Table 4.14: SynPCI card statistics for 518 bytes packets at 500 Mbit/s

Minimum delay	8.80 μs
Maximum delay	131.9 μs
Mean	24.43 μs
99^{th} percentile	39.30 μs
1^{st}-99^{th} interpercentile range	29.83 μs
Standard deviation	10.15 μs
Round Trip Time	18.26 μs

The distribution of the delay (without outliers) and the evolution of the maximum minimum and mean delay are shown in figures 4.28 and 4.29, where we can see the wider distribution of the observations. The shape of the distribution is due to the problems that the synPCI card has when the received packets rate is very high as we saw for the 64 bytes packets.

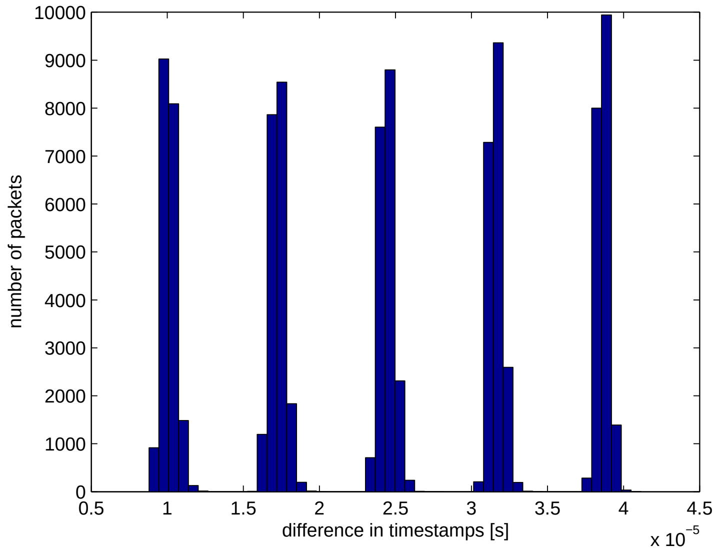

Figure 4.28: Histogram for 518 bytes packets at 500 Mbit/s captured by the synPCI card

SynPCI: 1418 bytes packets at 10 Mbit/s

The traffic generator sends a traffic burst consisting in 36000 periodic packets with a fixed length of 1418 bytes (1400 bytes IP packet + 18 bytes Ethernet

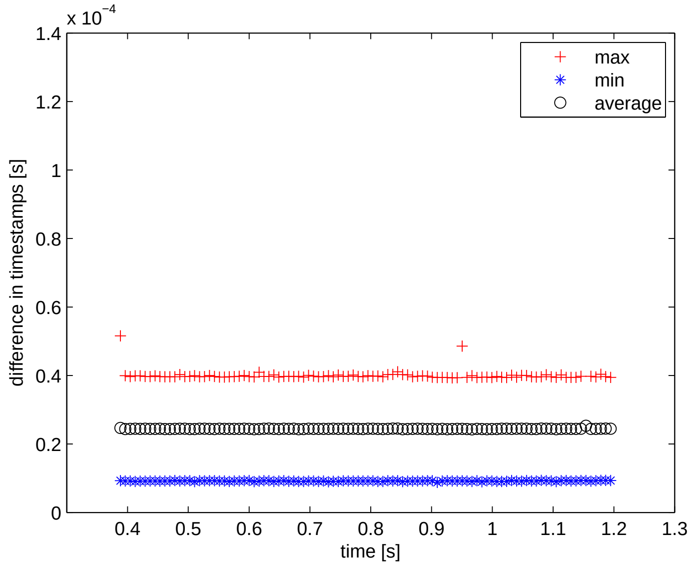

Figure 4.29: Delay evolution for 518 bytes packets at 500 Mbit/s captured by the synPCI card

header and trailer). The transmission bit rate is 10 Mbit/s (1% of the link capacity), the duration of the burst is 36.91 seconds and the packet interarrival time is 1.025 ms.

Table 4.15 shows the round trip time, mean delay and other statistical information. Again the bigger packets implies also a higher transmission delay, but the dispersion parameters are approximately the same or even lower.

Table 4.15: SynPCI card statistics for 1418 bytes packets at 10 Mbit/s

Minimum delay	23.47 μs
Maximum delay	83.85 μs
Mean	24.90 μs
99^{th} percentile	25.80 μs
1^{st}-99^{th} interpercentile range	1.83 μs
Standard deviation	700.7 ns
Round Trip Time	40.61 μs

The distribution of the delay (without outliers) and the evolution of the maximum minimum and mean delay are shown in figures 4.30 and 4.31.

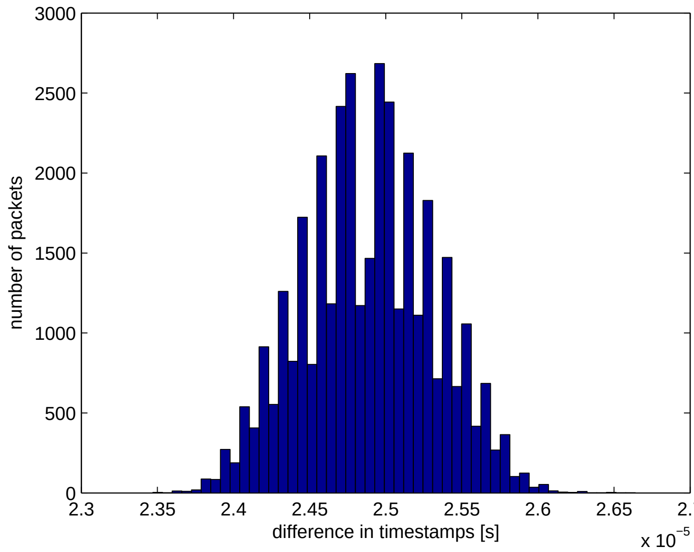

Figure 4.30: Histogram for 1418 bytes packets at 10 Mbit/s captured by the synPCI card

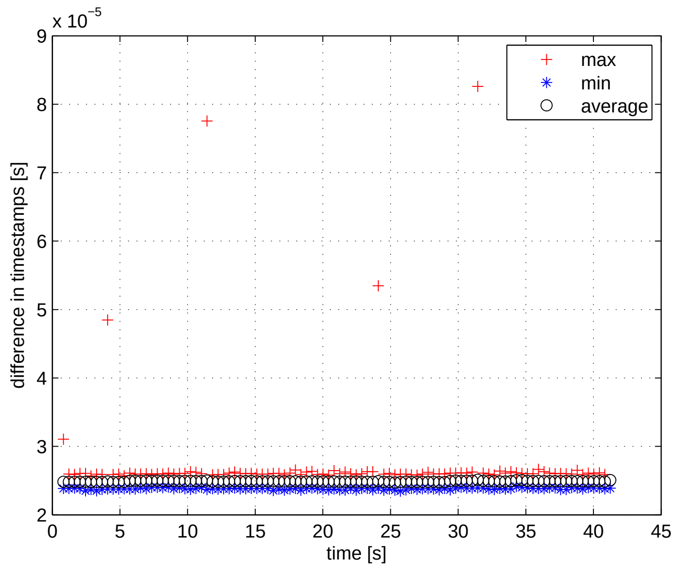

Figure 4.31: Delay evolution for 1418 bytes packets at 10 Mbit/s captured by the synPCI card

SynPCI: 1418 bytes packets at 200 Mbit/s

The traffic generator sends a traffic burst consisting in 36000 periodic packets with a fixed length of 1418 bytes (1400 bytes IP packet + 18 bytes Ethernet header and trailer). The transmission bit rate is 200 Mbit/s (20% of the link capacity), the duration of the burst is 1.845 seconds and the packet interarrival time is 51.25 μs.

Table 4.16 shows the round trip time, mean delay and other statistical information. The 99^{th} percentile and the standard deviation has raised due to the higher data transmission rate.

Table 4.16: SynPCI card statistics for 1418 bytes packets at 200 Mbit/s

Minimum delay	24.49 μs
Maximum delay	27.72 μs
Mean	26.49 μs
99^{th} percentile	27.35 μs
1^{st}-99^{th} interpercentile range	1.68 μs
Standard deviation	821.7 ns
Round Trip Time	40.61 μs

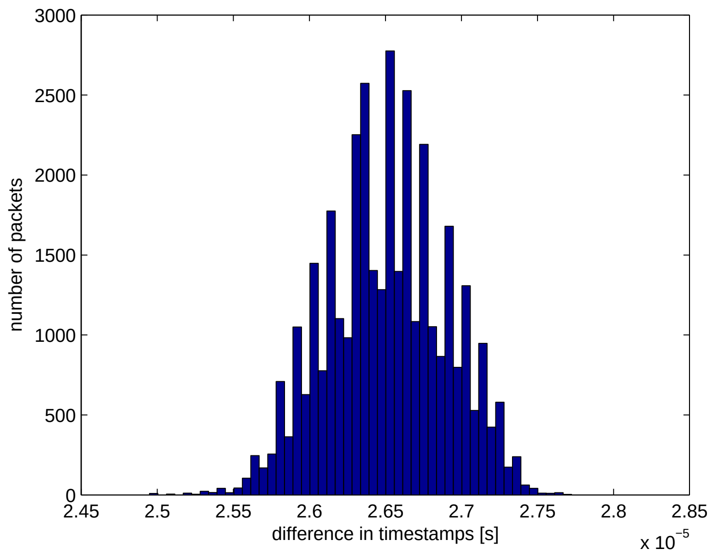

Figure 4.32: Histogram for 1418 bytes packets at 200 Mbit/s captured by the synPCI card

The distribution of the delay and the evolution of the maximum minimum and mean delay are shown in figures 4.32 and 4.33. In this case, by chance, there are not outliers in the capture.

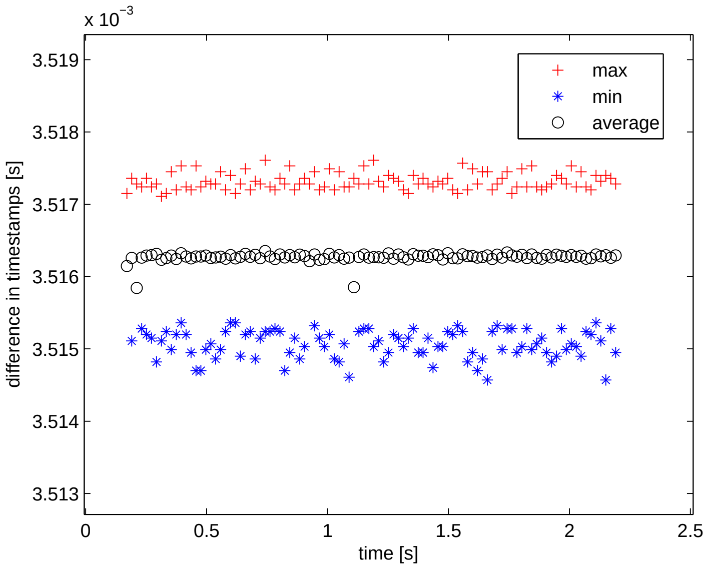

Figure 4.33: Delay evolution for 1418 bytes packets at 200 Mbit/s captured by the synPCI card

SynPCI: 1418 bytes packets at 500 Mbit/s

The traffic generator sends a traffic burst consisting in 36000 periodic packets with a fixed length of 1418 bytes (1400 bytes IP packet + 18 bytes Ethernet header and trailer). The transmission bit rate is 500 Mbit/s (50% of the link capacity), the duration of the burst is 0.738 seconds and the packet interarrival time is 20.50 μs.

Table 4.17: SynPCI card statistics for 1418 bytes packets at 500 Mbit/s

Minimum delay	25.01 μs
Maximum delay	134.6 μs
Mean	27.91 μs
99^{th} percentile	28.72 μs
1^{st}-99^{th} interpercentile range	1.42 μs
Standard deviation	899.1 ns
Round Trip Time	40.61 μs

Table 4.17 shows the round trip time, mean delay and other statistical information. At 500 Mbit/s the performance of the synPCI card is worse than for lower transmission rates, and therefore the 99^{th} percentile and the standard deviation have increased their values.

The distribution of the delay (without outliers) and the evolution of the maximum minimum and mean delay are shown in figures 4.34 and 4.35.

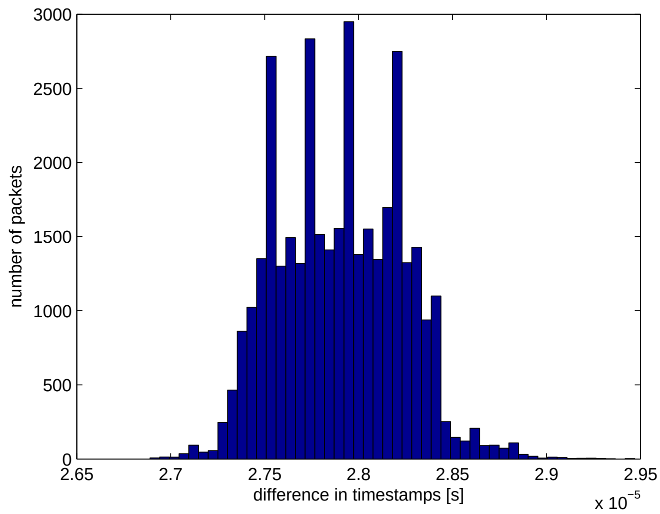

Figure 4.34: Histogram for 1418 bytes packets at 500 Mbit/s captured by the synPCI card

4.2.3 Results Summary

To summarise all the previous data, we have gathered the most important information in table 4.18.

Looking at the table, we can see that the mean and the 99^{th} percentile do not change significantly with the bit rate but they do with the packet size due to the increase in the round trip time as we will see in the next section. The dispersion parameters are also not affected by the traffic shape, only with high bit rates this parameters are smaller because of the short time of the capture.

The synPCI card, however, is considerably affected by the bit rate, and its performance is diminished at high transmission speeds. It also has an unstable behaviour when it receives a high amount of packets per seconds,

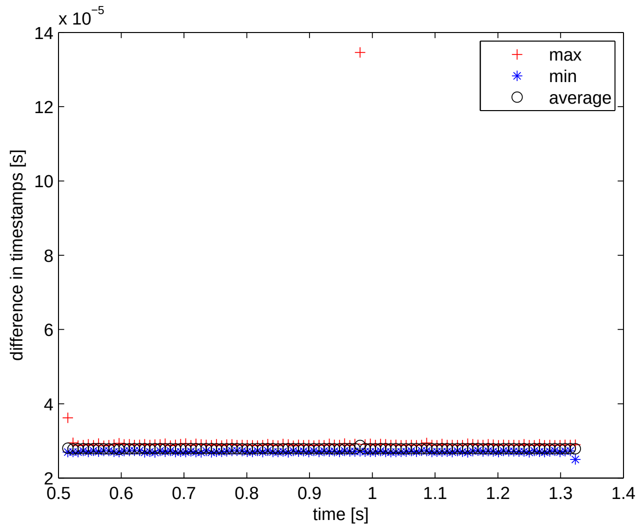

Figure 4.35: Delay evolution for 1418 bytes packets at 500 Mbit/s captured by the synPCI card

that is the reason why it seems to return better results when it capture bigger packets.

Comparing the two cards it is obvious that the DAG card provides a better performance and it has the advantage of not being affected by the transmission bit rate.

4.2.4 Round Trip Time and Packet Size

We have seen that the round trip time of the packets sent by the traffic generator depends on the size of the packets. Now we will see how it depends and what is the relationship between the two variables.

Table 4.19 shows the information that have been acquired at the laboratory for that purpose, including minimum, maximum and mean round trip time for different packet size. We have also checked that the round trip time does not change with the transmission rate.

This information is represented in figure 4.36, where it is easily identifiable the relation between the round trip time and the packet size. As we can see the relation is lineal with a slope of approximately 25 nanoseconds per byte. This delay is due to the conversion performed by the switch in both ways (optical-electrical and electrical-optical).

Table 4.18: Statistics summary for different bit rates and packet sizes

	Mean	99^{th} perc.	interprc range	Std deviation
DAG 64 B @ 10 Mbit/s	3.303 μs	3.384 μs	185 ns	39.45 ns
DAG 64 B @ 200 Mbit/s	3.291 μs	3.356 μs	129 ns	28.39 ns
DAG 64 B @ 500 Mbit/s	3.309 μs	3.370 μs	127 ns	27.89 ns
DAG 518 B @ 10 Mbit/s	8.731 μs	8.803 μs	148 ns	32.28 ns
DAG 518 B @ 200 Mbit/s	8.753 μs	8.813 μs	120 ns	26.91 ns
DAG 518 B @ 500 Mbit/s	8.755 μs	8.816 μs	121 ns	26.71 ns
DAG 1418 B @ 10 Mbit/s	20.160 μs	20.228 μs	139 ns	30.61 ns
DAG 1418 B @ 200 Mbit/s	20.108 μs	20.170 μs	125 ns	27.81 ns
DAG 1418 B @ 500 Mbit/s	19.900 μs	19.970 μs	122 ns	27.18 ns
SynPCI 64 B @ 10 Mbit/s	14.92 μs	15.97 μs	2.08 μs	736 ns
SynPCI 64 B @ 200 Mbit/s	12.265 μs	15.85 μs	7.25 μs	2.328 μs
SynPCI 64 B @ 500 Mbit/s	N/A	N/A	N/A	N/A
SynPCI 518 B @ 10 Mbit/s	24.346 μs	25.47 μs	2.17 μs	776.35 ns
SynPCI 518 B @ 200 Mbit/s	25.547 μs	26.39 μs	1.54 μs	823.2 ns
SynPCI 518 B @ 500 Mbit/s	24.43 μs	39.30 μs	29.83 μs	10.15 μs
SynPCI 1418 B @ 10 Mbit/s	24.90 μs	25.80 μs	1.83 μs	700.7 ns
SynPCI 1418 B @ 200 Mbit/s	26.49 μs	27.35 μs	1.68 μs	821.7 ns
SynPCI 1418 B @ 500 Mbit/s	27.91 μs	28.72 μs	1.42 μs	899.1 ns

4.2.5 Effects of unsynchronisation

Finally we are going to see the results of a capture where the DAG card is synchronised with the GPS signal but the traffic generator is out of sync. First of all, there will be an offset between the clocks of both devices due to the lack of synchronisation, but this problem could be fixed if we know beforehand what is the difference between the clocks. The worst problem is that there exist a time drift that causes this difference to be changing constantly. This drift can also change depending on the conditions of the unsynchronised oscillator (mostly the temperature).

Table 4.19: Results for Round Trip Time in relation to packet size

Packet size (Bytes)	46	300	500	700	900	1100	1300	1500
Mean RTT (μs)	7.24	13.24	18.26	23.24	28.24	33.24	38.23	43.11
Min RTT (μs)	6.97	12.97	17.98	22.97	27.97	32.97	37.97	42.51
Max RTT (μs)	7.52	13.52	18.52	23.52	28.52	33.52	38.48	43.48

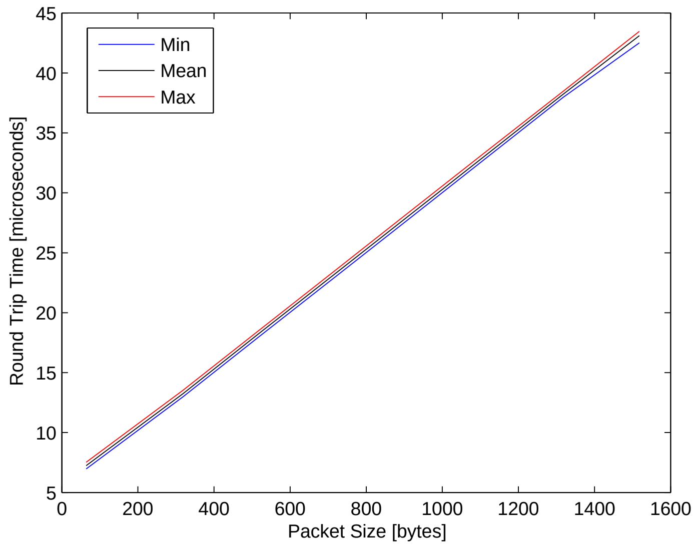

Figure 4.36: Relation between packet size and round trip time

Figure 4.37 shows the evolution by intervals performed on a capture of 64 bytes packets with a bit rate of 1 Mbit/s using the DAG card. We have eliminated the offset in the first observation supposing that both clocks were perfectly synchronised at the beginning of the capture. We can clearly see the drift of approximately 0.1 microsecond per second.

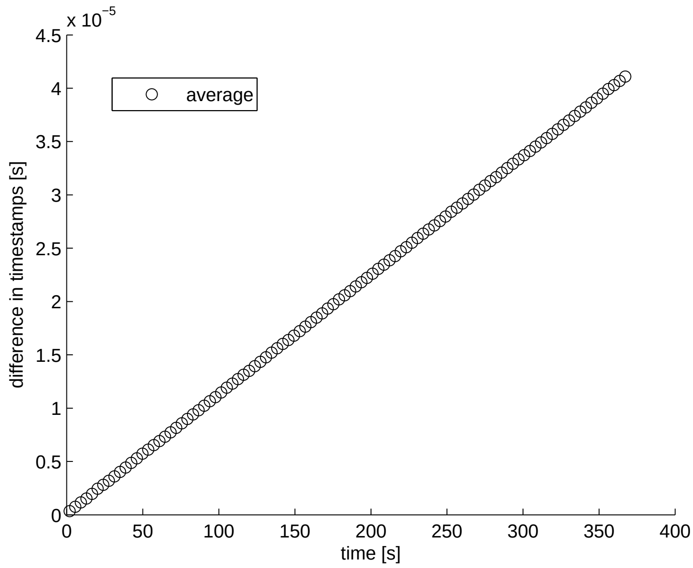

Figure 4.37: Time drift for unsynchronised capture

Chapter 5

Conclusions

Synchronisation over the network is a very important feature for telecommunications in general, but it becomes even more important when we deal with experimental research among different computers and networks where synchronisation is a major issue in order to achieve acceptable results.

In this thesis we analysed and discussed the performance of two different synchronisation devices. One commercial off-the-shelf capture card, and another card developed at Helsinki University of Technology by the Networking Laboratory department.

We have used GPS technology to get all the computers of the measurement setup in sync. Using the satellite signal as a time reference is equivalent to have an atomic precision clock inside the GPS receivers of the laboratory. This signal is then distributed to the devices where an accurate synchronisation is needed. But this is not the only advantage that GPS synchronisation provides, when working with multi-site networks, GPS allows us to synchronise all the networks no matters where are them or how far are from each other. We only need a GPS antenna and receiver in every location.

After the analysis we have seen than the performance of the synPCI card is not as good as the DAG card, but it still provides a better accuracy than a computer having the PPS synchronisation signal directly input to the serial port. The main advantage is that synPCI card avoids interrupt latencies, what is specially important when the computer is under heavy system load. Another remarkable feature is that it quickly gets in sync with the GPS source after it is rebooted. The optical fibre inputs provides the synPCI card with immunity to electromagnetic noises and lower attenuation. However, its most important problem is that its performance is significantly degraded when it works with dense traffic at high transmission rate and it does not work properly when it receives a high number of packets per second.

The DAG card has demonstrated to have a better performance than the synPCI card. It is able to provide accurate timestamps and its behaviour is not degraded when a high bit rate burst is applied to its input ports. The DAG card, although is a non-expensive commercial card, is still more expensive than the synPCI card. The longer time it takes to achieve synchronisation is another disadvantage of the DAG card with respect to the synPCI card.

Synchronisation is not only important in the receiving devices, but is equally important in the traffic sources and in all the network monitoring points. A lack of synchronisation in any point can degrade the results of the research. For wide network investigation where a high number of monitoring points are required, the scalability of the synchronisation devices becomes a fundamental issue. The cheaper price of the devices we have studied make them an affordable solution when a high number of computers need to be synchronised.

To summarise, in this thesis we have studied some synchronisation technologies that provides us with a better timing accuracy, multi-site synchronisation capabilities, all of that at an affordable price allowing the scalability of the system. With accurate results network research can focus more in the subject of the research than in inaccuracies and timing errors.

5.1 Future Work

Future work should consist on additional evaluation of SynPCI card in wider networks, such as multiple location networks. A patch for the tcpdump tool providing nanosecond resolution is necessary to allow us to know better the true performance of the card. Finally, it would be also useful to develop drivers for other operating system.

Bibliography

[1] BENNINGTON, J. *IEEE 1588 Precision Timing Protocol (PTP)*, Electronic Design. January 2007. http://eepn.com/locator/products/ArticleID/34270/34270.html.

[2] DANA, P. H. Global positioning system overview. Department of Geography, The University of Colorado at Boulder. http://www.colorado.edu/geography/gcraft/notes/gps/gps_f.html.

[3] ESHAM, B. D. http://commons.wikimedia.org/wiki/Image:Network_Time_Protocol_servers_and_clients.svg.

[4] HERNÁNDEZ, A., AND MAGAÑA, E. One-way delay measurement and characterization. Universidad Pública de Navarra, Pamplona, Spain, 2007.

[5] HOFMANN-WELLENHOF, B., LICHTENEGGER, H., AND COLLINS, J. *GPS Theory and Practice*. New York, 2001.

[6] IEC 61588 FIRST EDITION 2004-09. *IEEE 1588, Precision clock synchronization protocol for networked measurement and control systems*, 2004.

[7] KIHARA, M., ONO, S., AND ESKELINEN, P. *Digital Clocks for Synchronization and Communications*. Artech House, Inc., 2003.

[8] LANGLEY, R. The mathematics of gps. *GPS World* (July/August 1991).

[9] MARRISON, W. A. The evolution of the quartz crystal clock.

[10] PARKINSON, B. W., AND SPILKER, J. J. *Global Positioning System: Theory and Applications*. Aiaa, 1996.

[11] PEUHKURI, M., GRÖHN, A., AND SIMOLA, O. Clock synchronisation in industry-standard computer hardware. TKK Helsinki University of Technology.

[12] Press, Flannery, Tekolsky, and Vetterling. *Numerical Recipes, The Art of Scientific Computing.* Cambridge University Press, 1986.

[13] Range Commander's Council. Telecommunication and Timing Group. *IRIG Serial Time Code Formats*, 2004.

[14] Stefano Bregni. *Synchronization of Digital Telecommunications Networks.* John Wiley & Sons, Ltd., 2002.

[15] Vig, J. R. Introduction to quartz frequency standards. Army Research Laboratory, Electronics and Power Sources Directorate, October 1992. http://www.ieee-uffc.org/freqcontrol/quartz/vig/vigtoc.htm.