TIME SYNCHRONISATION OF MOBILE MAPPING SYSTEM : AN INNOVATIVE APPROACH

Sagnik Majumder, Raunaq Nandy Majumdar
Birla Institute of Technology and Science (BITS) Pilani, India
{f2014464,f2014396}@pilani.bits-pilani.ac.in

ABSTRACT

The difficulty that researchers working on Mobile Mapping System are currently facing in building their own Mobile Mapping System (MMS) is that the Terrestrial Laser Scanner (TLS) and the Global Positioning System (GPS) are from different manufacturers and cannot be easily time synchronised. Hence, geo-referencing of the acquired 3D scan with accuracy is not possible. In this particular work, RiVLib libraries have been used and executable files, to extract the time stamp information along with other scan attributes from the '.rxp' file produced by the TLS, have been built and, the GPS and the TLS are synchronised and a scan is intitiated by pre-setting the scan parameters. Servers were set up on a computer and could the TLS and the GPS were finally synchronized. The computer can act both as a server and a client. When it becomes a client, it takes the accurate time information from the GPS receiver using Network Time Protocol and upon becoming a server it passes on that time information to the TLS called Dynamic Host Configuration Protocol. Thus the TLS indirectly gets the time information from the GPS receiver and they are time synchronised.

LIST OF ABBREVIATIONS : ***TLS*** *– Terrestrial Laser Scanner,* ***GPS*** *– Global Positioning System,* ***IRNSS*** *- Indian Regional Navigation Satellite System,* ***NTP*** *– Network Time Protocol,* ***NTPD*** *– Network Time Protocol Daemon,* ***DHCP*** *– Direct Host Control Protocol,* ***GPSD*** *– Global Positioning System Daemon,* ***IMU*** *- Inertial Measurement Unit,* ***GUI*** *– Graphical User Interface,* ***NMEA*** *- National Marine Electronics Association,* ***HMI*** *– Human Machine Interface*

1. INTRODUCTION

1.1. Mobile Mapping System -

Mobile mapping is collecting geo-spatial data from a mobile device. It consists of mainly a laser scanner (2D/3D), a GPS device an Inertial Measurement Unit (IMU). The system has various real life applications like analysing the road condition, analysing the building material quality etc. The use of MMS in road condition surveying (roads, structures, facilities, exclusive use) reduces exclusive road use time in site operations, and enables the acquisition of 3D data for roads and their surroundings in an efficient yet low-cost manner. Through this technology, it is possible to gain an accurate and detailed understanding of the current situation. There are many other uses of the Mobile Mapping System that will be discussed in due course of time**.**

1.2. Laser Scanning -

Nowadays, terrestrial laser scanning has become a very efficient technique for geodetic applications. The use of laser scanners is continuously increasing. Different classes of laser scanners from several manufacturers are available. However, a strict classification of the laser scanners is quite difficult: on which point of view should the classification be based? Conceivable classifications concern the range or the point density or the point accuracy or the field of view.

1.3. Principles And Methods Of Laser Scanning -

Laser scanning means the deflection of a laser beam by moving (sweeping or rotating) mirrors, the reflection of the laser beam on object surfaces, and the receiving of the reflected laser beam. In opposite to measurements on reflectors, the accuracy of distance measurements depends on the intensity of the reflected laser beam. Physical laws describe the functionality between accuracy and intensity (Gerthsen 1993). Main parameters in these functions are the distance, the angle of incidence, and surface properties (Ingensand et al. 2003). For calibration of total stations, specific procedures were developed in the last years. Instrumental errors as well as angular resolutions and distance accuracies can be defined. Unfortunately, the mechanical design of laser scanners is different to total stations. Most investigations cannot be applied to laser scanners. The measurements in two faces as well as the repetition of single point measurements are impossible. Further, a complicating factor is that most of the manufactures of laser scanners have no experiences in how to construct "geodetic" instruments and how to minimize instrumental errors. This leads to the conclusion that laser scanners have to be investigated from the point of geodetic metrology. Mostly in ways which are optimised for the specific instrument and which are unstandardized.

In this work, Riegl VZ-400 has been used to serve the purpose (figure 1). It is a 3-D laser scanner that can be configured to use different modes of scanning. This work mainly involved scanning in line scan mode. It comes with an internal GPS and has provisions for an external GPS too. It also consists of a mountable battery, an LCD screen, a Nikon camera for image acquisition, GPS receiver, antennae and class 1 laser emitter. It comes with a tripod stand for mounting it for field purpose. It brings along with it a software called RiScan pro for viewing and processing the scanned data and acquiring other necessary details from the scans. The socket panel has an input port for PPS and NMEA signals (Trigger socket), two charging ports (figure 2), an ethernet port and an USB port.

Figure 1 – External View of TLS

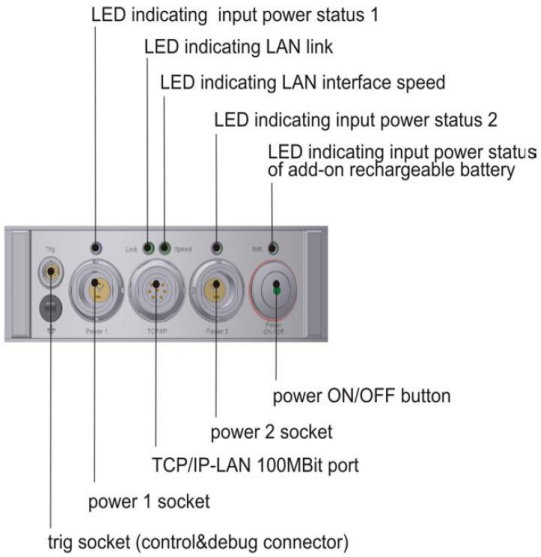

Figure 2 – Connection Port of TLS

1.4. GPS -

The Global Positioning System (GPS) is a space-based navigation system that provides location and time information in all weather conditions, anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites. The system provides critical capabilities to military, civil, and commercial users around the world. The United States government created the system, maintains it, and makes it freely accessible to anyone with a GPS receiver. The US began the GPS project in 1973 to overcome the limitations of previous navigation systems, integrating ideas from several predecessors, including a number of classified engineering design studies from the 1960s. The U.S. Department of Defence (DOD) developed the system, which originally used 24 satellites. It became fully operational in 1995. Roger L. Easton, Ivan A. Getting and Bradford Parkinson of the Applied Physics Laboratory are credited with inventing it. The process of fixing the position, velocity and altitude of a point is called as Trilateration. It generally uses 6 satellites for a precise fix and the accurate time provided by the satellites is maintained by a highly precise atomic clock on board the satellites which is further verified by a quartz crystal clock inside the GPS receiver.

An IRNSS/GPS/SBAS receiver, which is an indigenous GPS receiver made by Accord Software and Systems in India, has been used. It comes with a monitor (figure 3), a receiver, a battery charger, an AC-DC adapter, a DC-DC adapter, several input and output ports and different wires for connection.

Figure 3 – External View of an IRNSS Receiver

2. MAIN OBJECTIVES

2.1. EXTRACTION OF TIME STAMP INFORMATION OF TLS:

The time stamps or time information of the laser scans obtained directly from the '.rxp' file, produced by the TLS, using RiScan Pro showed anomalies in the sense that they were not increasing regularly in accordance with the order of scanning, they kept looping over and over again and they also did not start from zero value. So, as the first objective of this research, it was required to somehow extract the correct timestamps of the scanned points which increased in accordance with the order of scanning.

2.2. TIME SYNCHRONIZATION OF GPS AND TLS:

The other objective was to time synchronise the GPS and the TLS. The synchronisation could be either achieved by a software based approach or by a hardware based approach. The main impediment in synchronisation is that the manufacturers of the two systems and the data sampling rates of the two devices are different and can't be directly synchronised. But to geo reference the acquired scan data, it was necessary to time synchronise both the devices.

2.3. HARDWARE AND SOFTWARE USED

The following instruments and devices were used during the experiment

1. Riegl Terrestrial Laser Scanner(TLS) – VZ-400
2. IRNSS/GPS/SBAS Receiver – by Accord Software and Systems.
3. Inertial Measurement Unit (IMU) –by Microstrain

The software, used, are as follows

1. IRNSS-UR GUI - For logging and processing GPS data and also for changing receiver settings
2. RISCAN-Pro – For configuring the TLS and for processing data obtained from the TLS
3. Code::Blocks IDE – For writing C and C++ code
4. RivLib Libraries
5. CMake – For building the code into executable files
6. Linux OS and related packages – For building and running soft servers

The TLS comes with an internal GPS which is perfectly synchronised with the device itself. The TLS gives around 122000 measurements per second with an accuracy of 3mm and measurement range of 600m in distance.

The IRNSS receiver served the purpose as it is similar to a GPS receiver .The GPS receiver gives about 1-4 data samples per second.

For the MMS to be properly implemented, the scans obtained from the TLS should be geo-referenced and time stamped in accordance with the external GPS receiver because the internal GPS gives imprecise and inaccurate values and is most likely to jeopardize the accuracy necessary for a highly efficient MMS.

The devices are physically disconnected and their data sampling rates are also different. There is an option in the TLS to connect an external GPS receiver to it and synchronise but it turned out that synchronisation couldn't be achieved even after following the instructions precisely. So it was required to time-synchronise the two devices in order to extract the accurate geo-referenced data for using it in building an efficient and precise MMS.

3. SOFTWARE BASED APPROACH FOR TIME SYNCHRONISATION

3.1. SOLUTION TO OUR PROBLEMS:

Ideas both for timestamp extraction and time synchronization of the GPS Receiver with the TLS were conceived.

The following steps were followed -

i) Setting up of a server on a PC which will connect to both scanner and GPS Receiver was required. The decision to feed the PC with the time from the GPS and feed the time to the TLS was taken. This seemed to be a credible and accurate approach for time synchronisation because GPS time is known to be very precise, precise to ten-sixth of a second as the time is maintained by crystal and atomic clocks.

ii) For time stamp extraction the decision to write our own code to extract the timestamps in a sorted order same as that of the points scanned, as opposed to the timestamps exported by RiScan-Pro in the form of a text file which were completely out of order, was made. The main impediment in this approach was that the '.rxp' was a very hidden and editor selective type of format which refused to open in any other editor. So, extracting data from such a file format seemed to be quite difficult.

3.2. EXTRACTION OF TIMESTAMPS FROM '.rxp' FILE AND AUTOMATIC INITIATION OF SCANS USING C CODE:

Two pieces of code were written:

i) A C code for time stamp extraction both from a '.rxp' file and live data streaming from the scanner.

ii) A C++ code for automatically initiating a scan and setting initial parameters for the scan.

3.2.1. Code #1 – C Code for Scan Attributes Extraction

The first code was a C code and was meant for timestamp extraction along with other attributes like local coordinates of the points scanned , the pps sync bit of the points and the 'end of frame' bit. The code mainly used the 'scanifc.h' header file. The code self-builds a text file by the same name as the rxp file but with a '.txt' extension in place of '.rxp' and writes all these parameters onto the text file in the order of their scanning. The timestamps obtained on the text file are also supposed to be in increasing order, same as that of scanning.

3.2.1.1. CODE IN APPENDIX

Finally the RiVLib library version 'rivlib-2_3_0-x86_64-linux-gcc44' was used for building the code on Ubuntu. The code was built into an 'attributes_output.exe' file for Ubuntu. It is also essential to put the file 'scanifc.dll' in a directory accessible by the code, preferably the same directory as the code.

3.2.1.2. Building and Running the Code on Ubuntu

To build the code on Ubuntu from the command line terminal, the following changes and installations were done –

No significant changes are required , if any modifications are needed for the particular system on which the code is being built , it is advisable that they are done in accordance with official Ubuntu documentation.

After that, the following commands were executed from the terminal window-

g++ -g -std=c++11 attributes_output_linux.c -o attributes_output_linux-I/home/user/rivlib-2_3_0-x86_64-linux-gcc44/include -L /home/user/rivlib-2_3_0-x86_64-linux-gcc44/lib -lscanifc-mt -lscanlib-mt

3.2.1.3. Output of the code

Finally the code was tested on laser scans to see if the code produced proper output in the form of a text file. Tests, both on old scans taken in Panorama or Line Scan mode and new scans, were conducted. The code worked perfectly on all the scans and produced output having information about the local coordinates of each point , their timestamp, the value of their pps sync bits, the value of their end of frame bits.

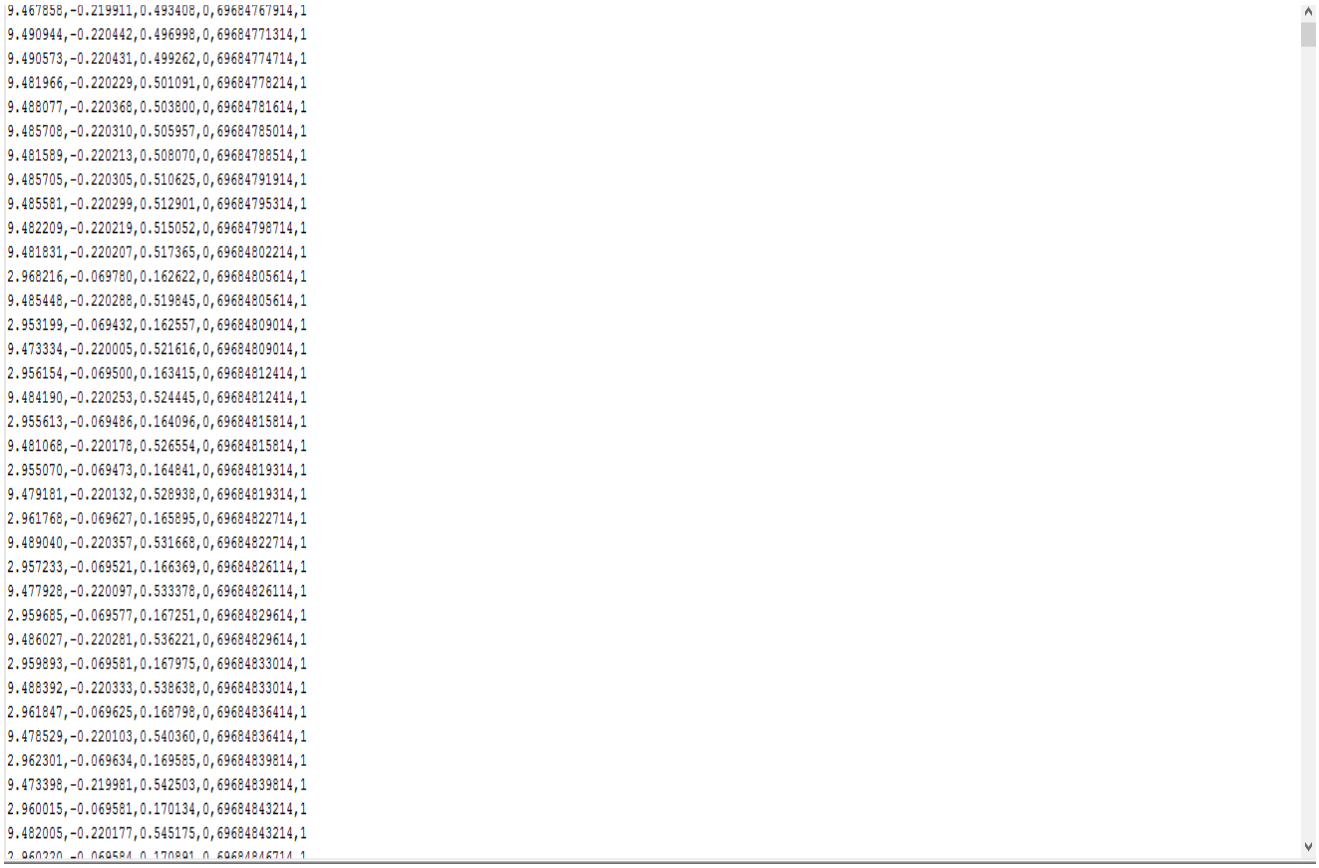

Figure 4 – Parts of the Text File Produced From '.rxp' File

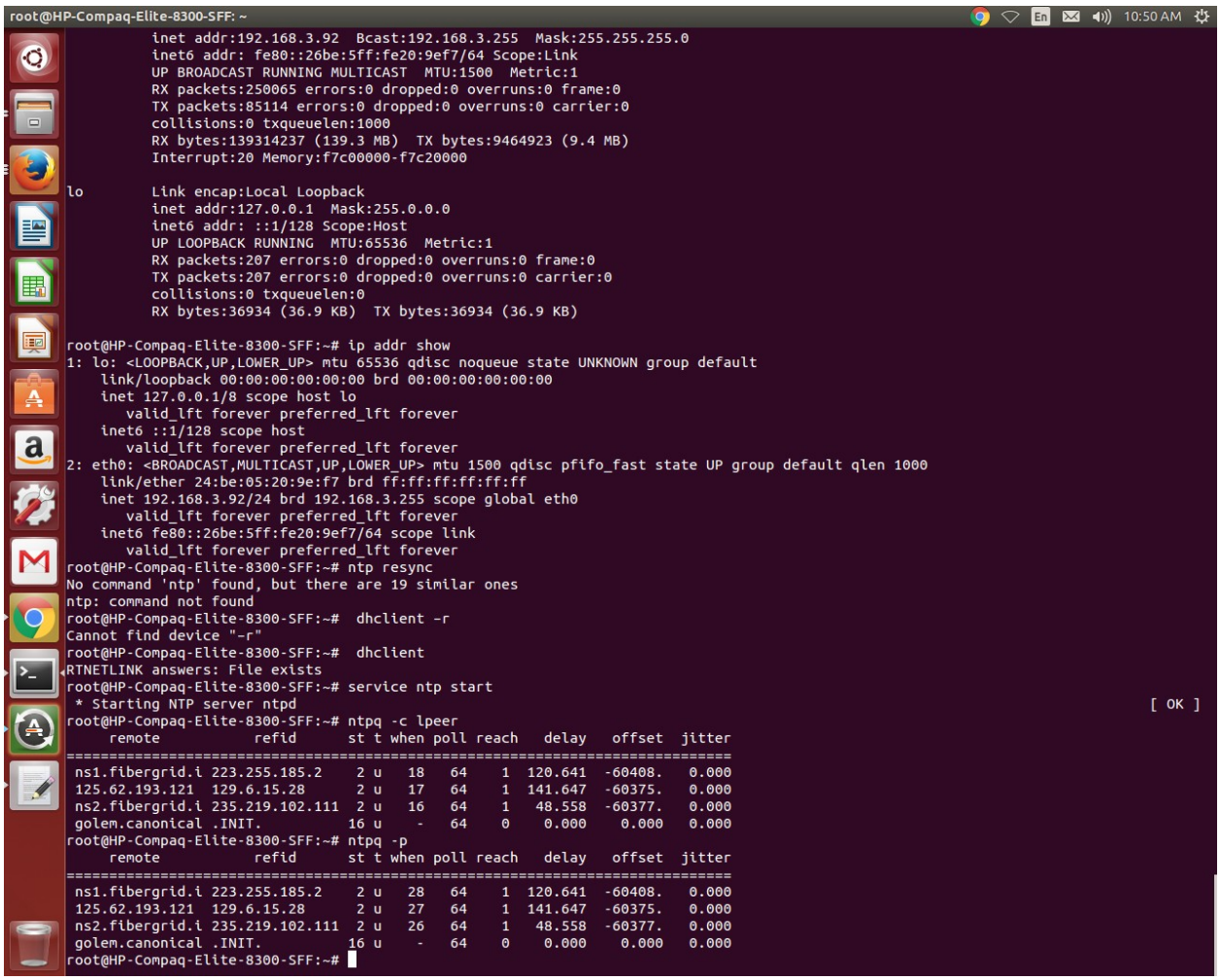

Figure 5 - 'ntpq -p' on Command Window

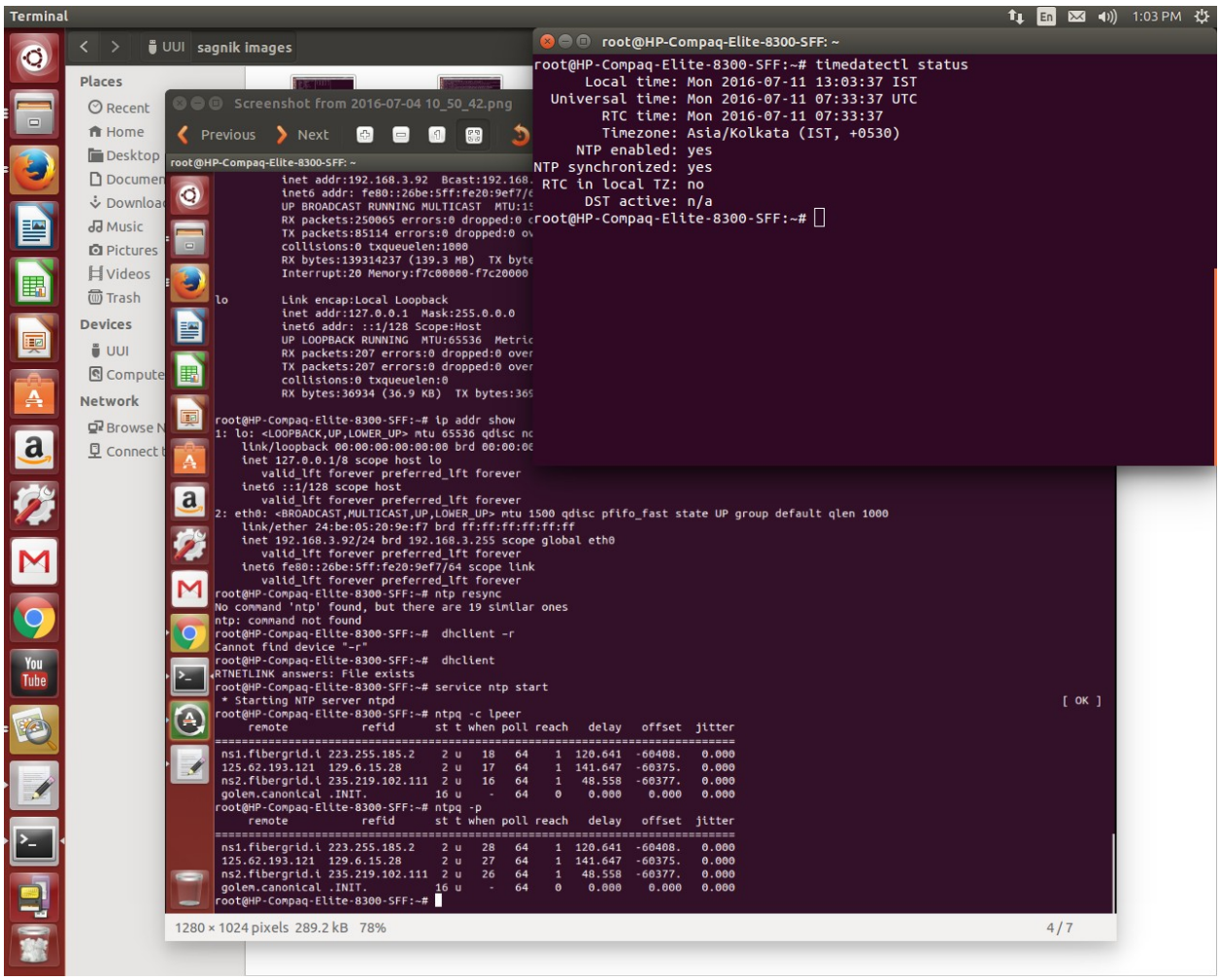

Figure 6 – 'timedatectl status' on Command Window

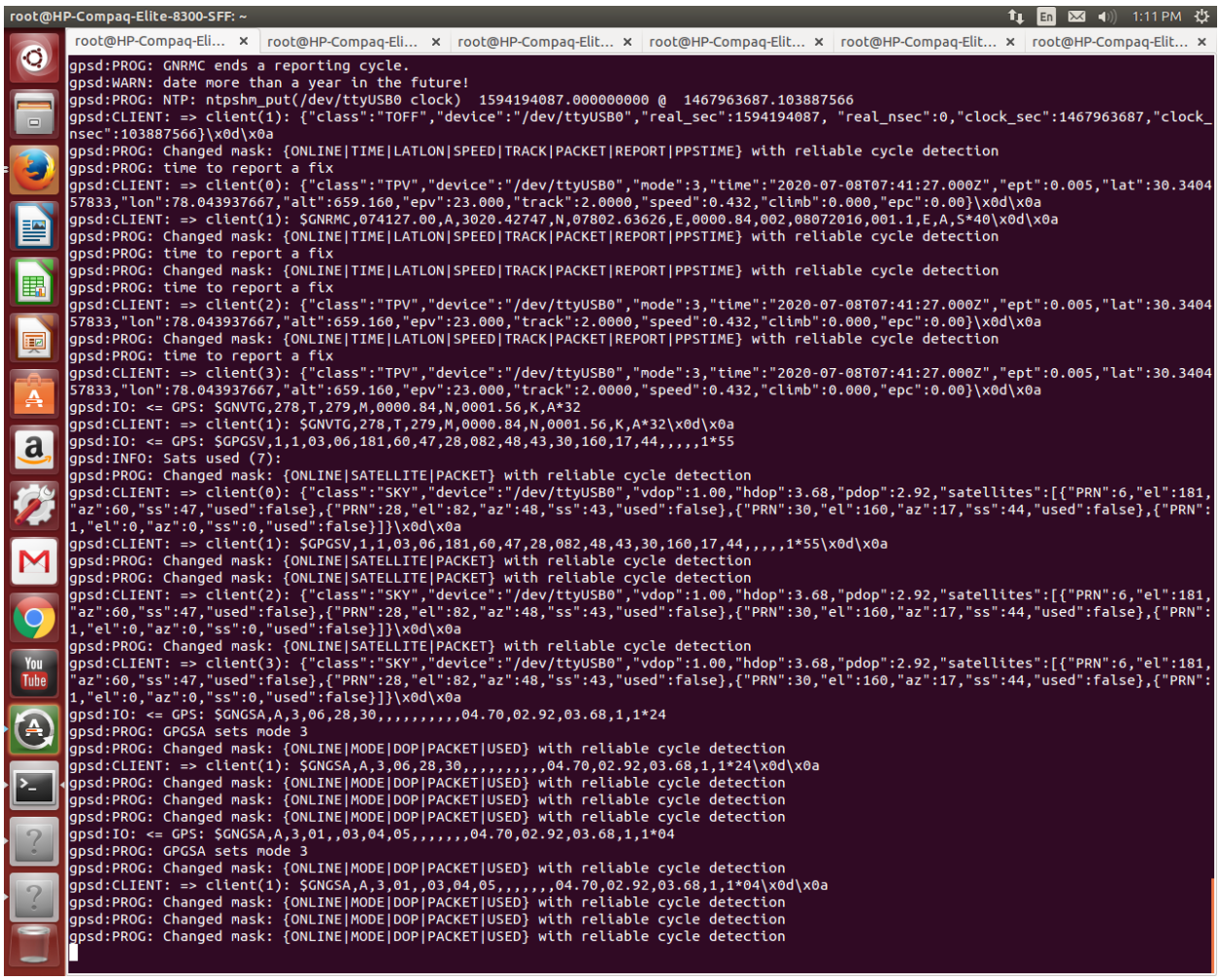

Figure 7 – '/usr/local/sbin/gpsd -n –N –D5/dev/ttyUSB0' on Command Window

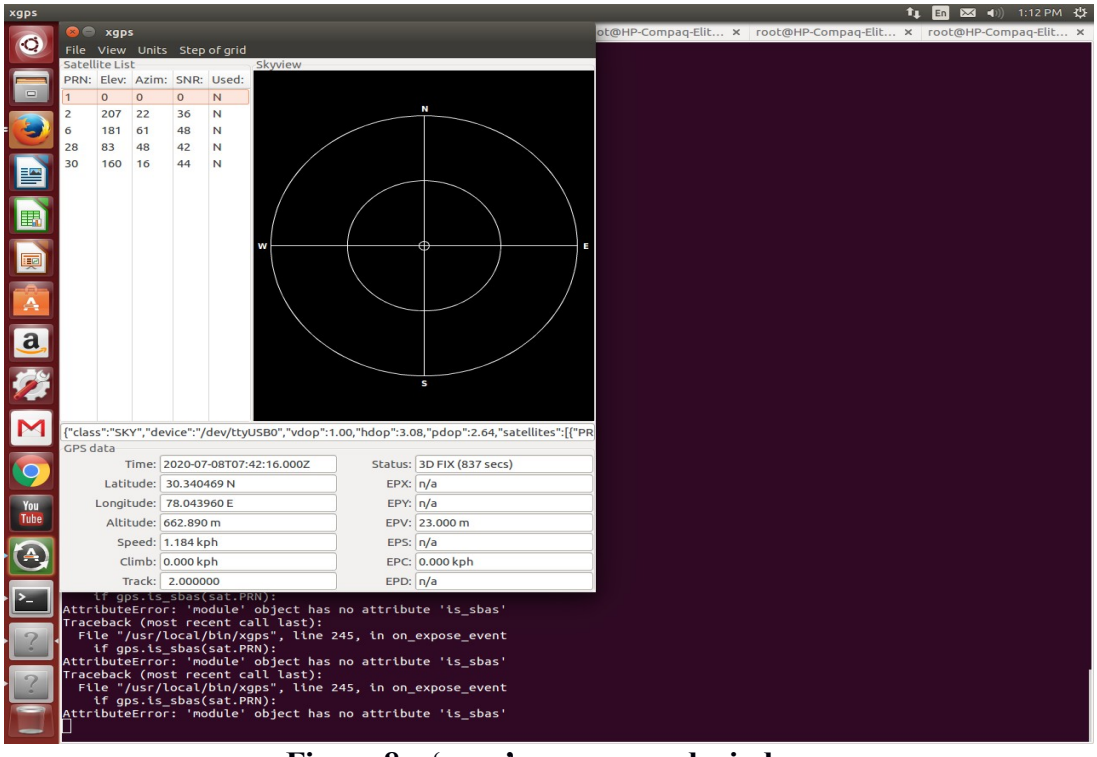

Figure 8 – 'xgps' on command window

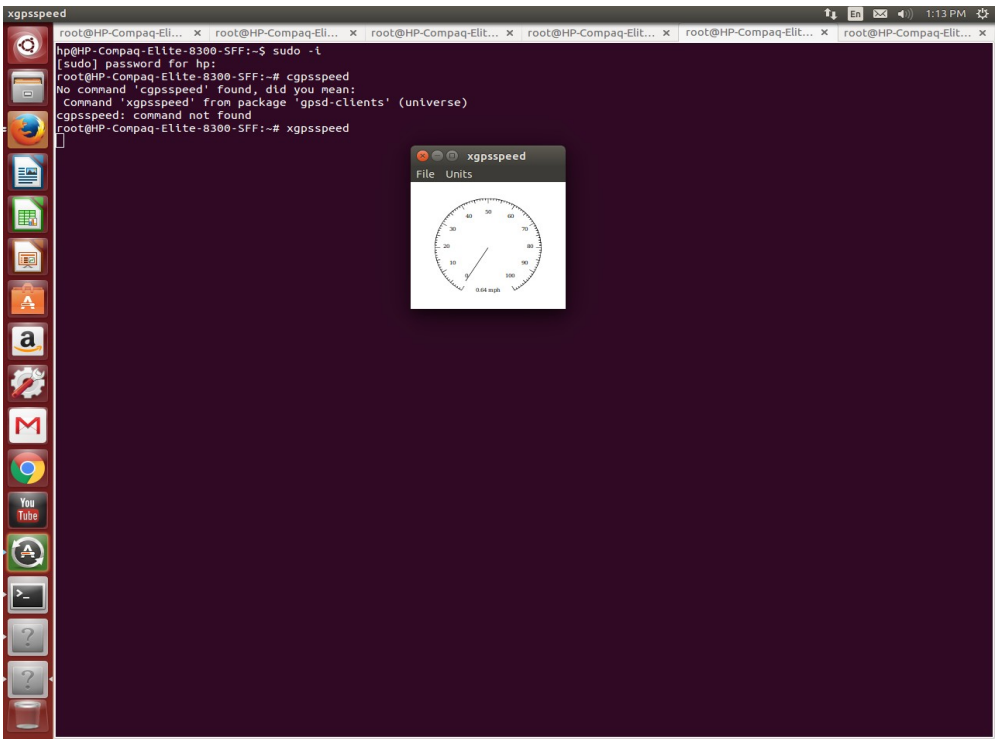

Figure 9 – 'xgpsspeed' on command window

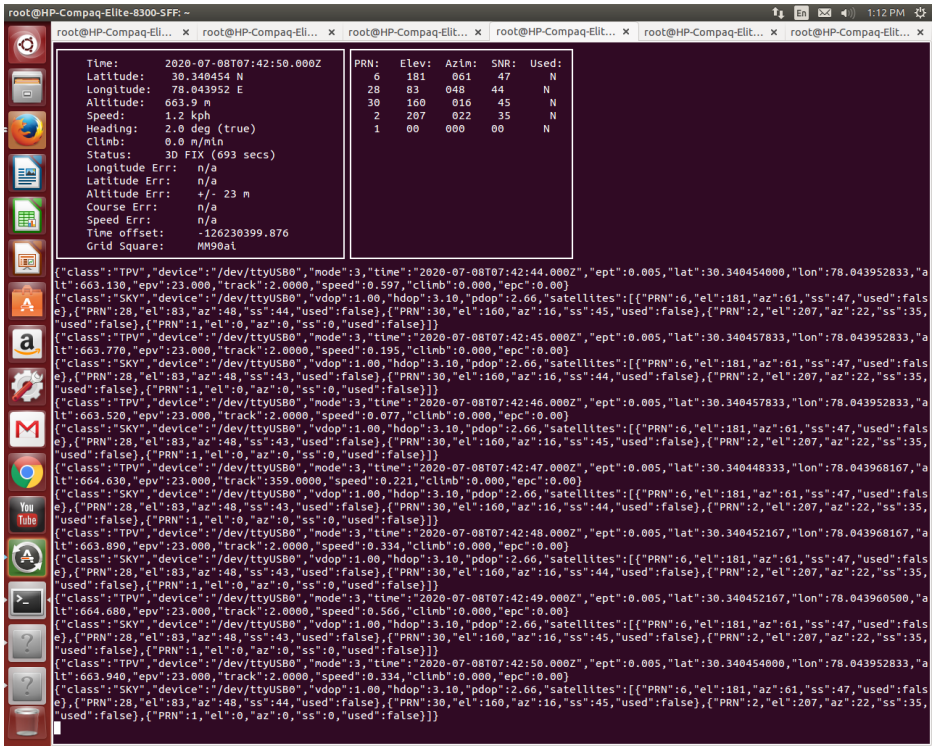

Figure 10 – 'cgps' on command window

At the end of the file the total number of points scanned is getting printed. It was also noticed that the difference between two consecutive points in the text file is 3400 and all the timestamps are arranged in increasing order of magnitude. Though the magnitude of the timestamp is peculiar because of its 15 bit decimal value and it is not the epoch or UNIX time as measured from January 1st , 1970, without the leap seconds. Another noticeable characteristic in the text file was that there was a difference of 3400 units between 2 consecutive points which showed that it was directly related to the time rate of scanning of the points. It also highlighted the fact that the points were being scanned by the scanner at a constant frequency. But both the timestamps and the

time difference between two consecutive points was somewhat large in magnitude which may be because of some internal factor of multiplication by the scanner.

3.2.2. Code #2 - Code to Synchronise With GPS, Initiate and Continue Scan until the Data Limit Is Reached

The second code was written so as to automatically terminate a previous scan and start a new scan only if the scanner whose identity given as command line argument is connected to the system(computer in this case).

The program forces the scanner to abort the present scan and start a new scan with the range horizontal scan angle range set to 180° and the vertical angle range set to (30° to 130°) with a step increment of 0.9875°. It also sets various parameters related synchronisation

i) The Talker ID of NMEA Communication Protocol or GPS External Format (GPZDA in this case),
ii) The Baud Rate or External Communication Baud Rate (115200 bps in this case),
iii) GPS Mode (set to External GPS via RS232 in this case),
iv) GPS data sequence or GPS External Sequence (set to PPS FIRST in this case) ,
v) GPS External Edge (set to positive in this case).

The scan data is logged to a '.rxp' file given in the second command line argument. The first command line argument gives the URI (Uniform Resource Identifier) of the scanner in the form of **<host_name_or_ip_address>:<service_name_or_port_number>.**

Examples -

"192.168.0.234", "192.168.0.234:20002",
"s9991234" or "s9991234:20002"

Before the scan starts, the internal storage is enabled and a connection between the scanner and the system is established. Once the pre-set data limit is reached, the scan is terminated by the program and the connection is closed. The data is logged onto a binary file and finally stored as a '.rxp' file.

3.2.2.1. CODE IN APPENDIX

3.2.2.2. Building the Code on Ubuntu

To build the code on Ubuntu from the command line terminal, make the following changes and installations –

> No significant changes are required , if any modifications are needed for the particular system on which the code is being built , it is advisable that they are done in accordance with the official Ubuntu documentation.

After that, execute the following commands from the terminal window-

> **g++ -g -std=c++11 gps_time_sync_2.cpp -o gps_time_sync_2 -I/home/user/rivlib-2_3_0-**

x86_64-linux-gcc44/include -L/home/user/rivlib-2_3_0-x86_64-linux-gcc44/lib -lscanifc-mt -lscanlib-mt -lctrllib-mt -lpthread

3.2.2.3. Running the Executable on Ubuntu

To run the executable on Ubuntu from the command line terminal, make the following changes and installations -

No significant changes are required , if any modifications are needed for the particular system on which the code is being built , it is advisable that they are done in accordance with official Ubuntu documentation.

After that, execute the following commands from the terminal window-

To run the code , the name of the executable is to be mentioned followed by the URI (Uniform Resource Identifier) of the scanner having the form **<host_name_or_ip_address>:<service_name_or_port_number>** and the name of '.rxp' file where the data is to be logged.

./gps_time_sync_2 <host_name_or_ip_address>:<service_name_or_port_number><filename>

Examples-
<host_name_or_ip_address>:<service_name_or_port_number>:
"192.168.0.234" OR "192.168.0.234:20002",
"s9991234" OR "s9991234:20002"
<filename>:"scan1.rxp"

3.2.2.4. Output of Code

The scan data is logged on to the '.rxp' file the name of which has been given as a command line argument.

3.2.3. Code #3 – Combined Code to Synchronise with GPS, Initiate and Continue Scan until the Data Limit Is Reached and Extract the Scan Attributes From The '.rxp' File or Live Data Stream

This combined code was written so that it singlehandedly performs the twin tasks of the previously mentioned C and C++ code. It first synchronizes the GPS with the TLS, terminates a previously running scan, initiates a new scan and completes the scan, finally logging the data onto an '.rxp' file. After that, the later part of the code extracts the attributes from the previously formed '.rxp' file.The program is wriiten in the form of C file aimed at extracting the scan attribute data from a '.rxp' file which uses a C++ code in the form of a header file which actually synchronises with the GPS, initiates a scan, completes it and logs a data to '.rxp' file to be used by the C code using the header.

3.2.3.1. CODE IN APPENDIX

3.2.3.2. Building the Code on Ubuntu

To build the code on Ubuntu from the command line terminal, make the following changes and installations –

No significant changes are required, if any modifications are needed for the particular system on which the code is being built, it is advisable that they are done in accordance with official Ubuntu documentation.

After that, execute the following commands from the terminal window-

g++ -g -std=c++11 attributes_output_edited_final.c -o attr_output_edited_final -I/home/user/rivlib-2_3_0-x86_64-linux-gcc44/include -L/home/user/rivlib-2_3_0-x86_64-linux-gcc44/lib -lscanifc-mt -lscanlib-mt -lctrllib-mt –lpthread

3.2.3.3. Running the Executable on Ubuntu

To run the executable on Ubuntu from the command line terminal, make the following changes and installations -

No significant changes are required, if any modifications are needed for the particular system on which the code is being built, it is advisable that they are done in accordance with official Ubuntu documentation.

After that, execute the following commands from the terminal window-

./attributes_output_edited_final <filename>
<host_name_or_ip_address>:<service_name_or_port_number> <filename>

Examples-
<host_name_or_ip_address>:<service_name_or_port_number>:
"192.168.0.234" OR "192.168.0.234:20002",
"s9991234" OR "s9991234:20002"
<filename>:“scan1.rxp”

3.2.3.4. Output of the Code

A ‘.rxp file’ is created and the scan attributes are extracted from the same ‘.rxp file’ onto a ‘.txt file’.

3.3. USE OF SERVERS TO TIME SYNCHRONISE TLS AND GPS RECEIVER:

This approach was adopted mainly because there were problems with synchronising by directly connecting the TLS and the Global Positioning System (GPS) receiver. So the decision of indirectly connecting the two instruments with a computer in between them which will act as an interfacing system for the devices was taken. The basic idea was that the computer will take the accurate time information from the GPS receiver and pass it on to the TLS so that the TLS can use the same to time-synchronise its scans with the GPS receiver. For that it was planned to use the computer both as a server and a client so that it can take time information from the receiver and provide it to the scanner. The property of transitivity was used in our approach in the sense that if the GPS receiver (denoted by A) is synchronised with the computer (denoted by B) and the computer (B) is synchronised with the TLS (denoted by C), then the receiver (A) will be synchronised with the TLS (C) .In this case, the computer was used as a client when it is communicating with the receiver and as a server. when it is communicating with the TLS. The process is illustrated below using an arrow diagram (figure 15).

Two types of communication protocol were used for this –

i) Network Time Protocol (NTP) –

NTP is a TCP/IP protocol for synchronising time over a network .A client requests the current time from a server and uses it to set its own clock. We needed NTP because we wanted to keep our computer time synchronised with the time source, the GPS receiver, in this case. For synchronisation with the time source, system supporting NTP communication uses NTP Daemon Tools (NTPD) which can act both as server and client. Even our personal computers can act as NTP clients using NTPD and connect with the NTP servers available all over the world to maintain an accurate time. The computer basically uses NTPD to take time from a source but in this case NTPD alone was not enough to obtain the time information to obtain the time interfacing. It needed another interfacing tool called GPS Daemon (GPSD) tool for communicating with the receiver.

ii) Dynamic Host Configuration Protocol (DHCP)-

The main reason behind using DHCP was that using the Human Machine Interface (HMI) or the control panel provided on the scanner we found that the scanner can be configured both as DHCP client and DHCP server. DHCP is used for communication between the computer and the TLS in which the computer (a DHCP server) will supply the time information to the TLS (a DHCP client). The TLS will use that time information to time-synchronise its scans with the GPS receiver.

The computer is an NTP client and takes the accurate time data from the GPS receiver which is an NTP server, via NTPD which then uses GPSD as an NTPD can’t directly communicate with GPS receiver and needs GPSD for this purpose. The computer then acts like a DHCP client and supplies the time information to the TLS which becomes a DHCP client which finally uses this time information to time synchronise its scans. In this way, the time synchronisation between the GPS receiver and the scanner was finally achieved. We had to work on it for several days as it tool time to understand the protocols and their functionalities well enough to use the daemon tools and set up

and configure the servers.

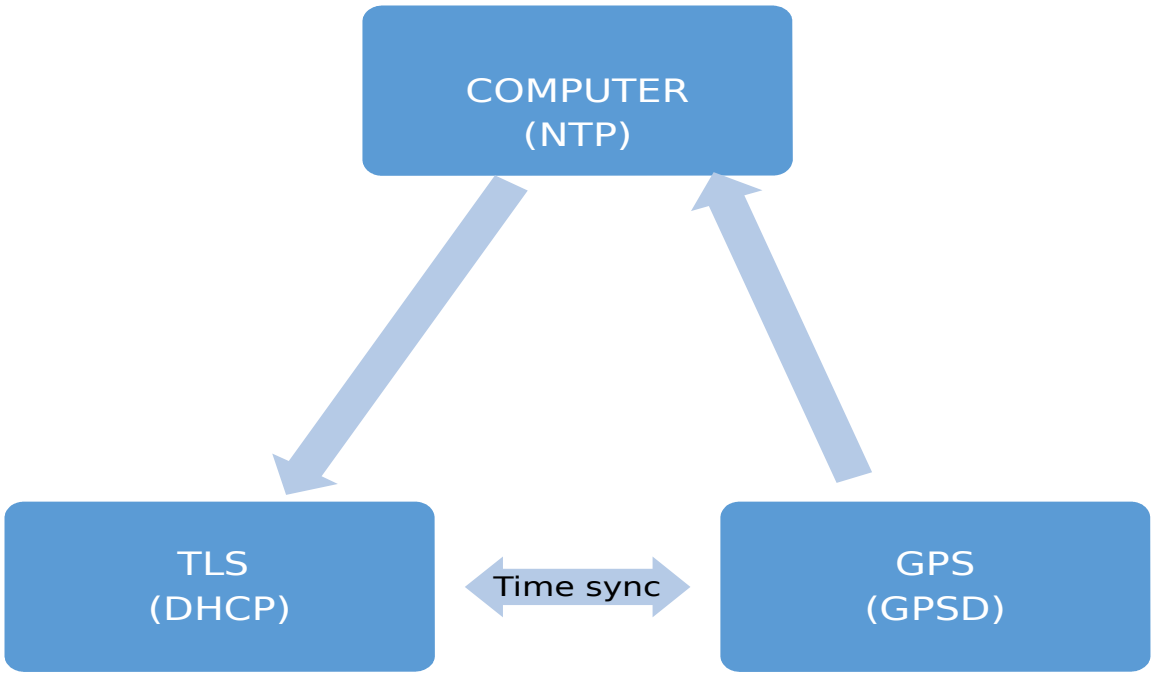

Figure 11 – Representative Diagram of Synchronisation

We have to set three servers and the procedures are as follows:

3.3.1. Using NTPD to set up NTP Communication

NTPD is a daemon tool which helps a system keep its internal clock synchronized with a time source. The time sources are basically NTP servers located all around the world.

NTP uses a hierarchical semi-layered system of time resources. Each level of this hierarchy is called a 'stratum'. A server synchronised to a stratum 'n' server will be running at stratum 'n+1'. It is the distance between the reference clock and the NTP server/client in terms of the number of intermediate servers/clients and it is so implemented to prevent cyclic dependencies. The lower the stratum number of the NTP server /client is, more accurate is the time kept by it. Stratum is not always an indication of quality or reliability; it is common to find stratum 3 time sources that are higher quality than other stratum 2 time sources. Telecommunication systems use a different definition for clock strata. A brief description of strata 0, 1, 2 and 3 is provided below (figure 16)

The NTPD tool understands the Network Time Protocol and can work both as a client and server in an NTP network. Besides that it can synchronize with local time sources. The NTPD cannot interface with GPS devices directly, but it has defined a number of interfaces which can be used by other daemons to interface GPS with NTPD.

The NTPD is either already installed on the system, or it can be downloaded from one of the package repositories. Linux distributions like Debian , Ubuntu and RedHat all have NTPD in their standard repository.

3.3.1.1. "Steps for Installing NTPD and Setting up NTP Server/Client on Ubuntu" in Appendix

3.3.1.1.2. “Running NTPD as Server/Client” in Appendix

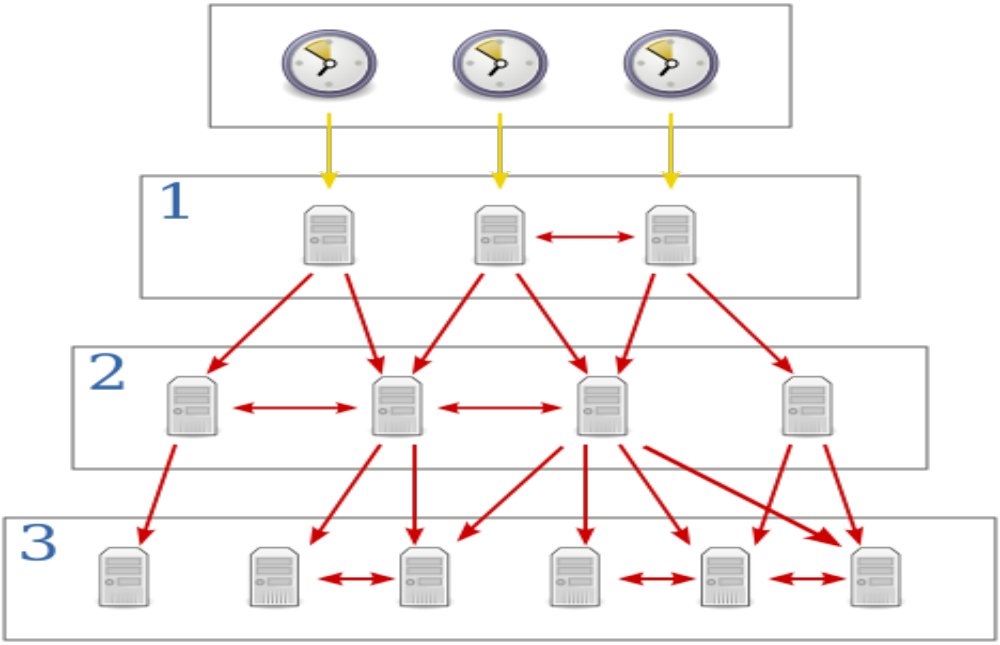

Figure 12 - Representative Diagram of NTP Server Time Synchronisation

3.3.2. Using GPSD to set up Communication between GPS Receiver and NTPD

NTPD can't communicate directly with a GPS device, so a separate daemon tool is needed for the GPS communication. As always in Linux there is more than one solution to this problem, but we have used the GPSD tool for this task. The GPSD tool is the most versatile of all GPS daemons currently available. It does not only interface well with most GPS models available, but it also communicates nicely with NTPD and has a special interface to make it possible to easily integrate in web pages etc. The GPSD tool setup package can be downloaded from **‘gpsd.berlios.de’**.

3.3.2.1 “Steps for Installing GPSD and Running It” in Appendix

3.3.2.1.2 “Running GPSD” in Appendix

3.3.3. Using DHCPD to set up DHCP Communication

During the initial days of our project , we found out that the TLS can act both as a DHCP server and as a DHCP client. So, we figured out that if we can somehow set up a DHCP server on our computer we can send the time information obtained by the computer from the GPS receiver to the TLS. This will definitely help us synchronise the TLS with the GPS receiver.

DHCP stands for Dynamic Host Configuration Protocol.It is a standardized network protocol used in Internet Protocol(IP) networks for dynamically distributing network configuration parameters such as IP addresses and networking parameters like Subnet Mask, default Gateway automatically from a DHCP server , reducing the need for a network administrator or a user to configure these settings manually. DHCP also helps to send the NTP server information from the DHCP server to the DHCP client.

3.3.3.1. “Steps for Installing GPSD and Running It” in Appendix

A piece of code was added to the ‘dhcpd.conf” file and both the actual version and the modified version of the file are shown below.

Even the '/etc/network/interfaces' and '/etc/NetworkManager/NetworkManager.conf' were modified as per requirements.

Both the original and changed version of the files are shown below.

The 'etc/dhcp/dhcpd.conf' file gives the subnet address, subnet mask,optional routers and optional broadcast address of the server . It also gives the range of IP addresses that can be leased by the dhcp server to the client.The range in this case is '192.168.3.100' to '192.168.3.200' .The default lease time of 600 seconds and the maximum lease time of 7200 seconds are also specified.

The subnet address is basically calculated by bitwise anding between the IP address and the Subnet Mask of the server. The IP address in this case is '192.168.3.xx' and the Subnet Mask is '255.255.255.0' .So the Subnet Address as calculated is '192.168.3.0'.

3.3.3.1.2. "Running the DHCP Server" in Appendix

In this way, the DHCP server was set up. Once both GPSD and NTPD are running and the DHCP communication has been established, the DHCP server on the computer sends the NTP server information on the computer to the DHCP client.

So, the computer receives time data from the GPS receiver via NTPD tool which indirectly communicates with the GPS using GPSD tool. The time information is then sent by the DHCP server on the computer to the TLS which is a DHCP client. Hence, the GPS receiver and the TLS are synchronised.

4. RESULTS

Time extraction and time synchronisation was finally possible with the help of our ingeniously written C and C++ code and shell scripts on Ubuntu. Our results are shown in Figure 5 – 10.

5. WORK TO BE DONE FOR FUTURE DEVELOPMENT

The time synchronisation of the TLS and the GPS receiver has been achieved. But since they are being synchronised indirectly without direct hard-connection, the HMI of the TLS doesn't show the 'Synchronisation : OK' message. So further study can be done on how to configure the servers and the TLS so the above mentioned message gets displayed upon synchronisation.

The previously mentioned executable files can extract different scan parameters like the spatial coordinates and the angular coordinates . These data from the text file created after running the executable files can be used to reconstruct 3D point clouds on any cloud processing software like CloudCompare and the generated point clouds can be used in many fields and applications like disaster management, fault management, analysis of constituent materials of structures and buildings, and others.

This work enables the transfer of only the time information from the GPS receiver to the TLS whereas for proper geo-referencing it is also required to transfer the fix data from the GPS receiver to the TLS. Though the fix data of the GPS is getting transmitted to the computer, it still remains to be seen whether the same data can also be transferred to the TLS for geo-referencing.

6. CONCLUSION

The time synchronisation of the Terrestrial Laser Scanner and the Global Positioning System has thus been achieved using the software based approach as discussed above. We were successful in our attempt to extract the time stamps and other necessary scan attributes from the scans and we could also automate the initiation of scanning by the TLS with pre-set parameters by using ingeniously built executable files. By following the above mentioned approach, a mobile mapping system consisting of TLS, GPS (in our case it is IRNSS receiver) and an IMU can be set up. Along with that, a mobile platform which can either be a trolley for lower order application or car for higher order application will be required. A laptop with internet connection for setting up the servers is also required. Thus, we can synchronise the devices and get a complete geo referenced scan and the time stamp information along with other scan attributes generated with the help of the RivLib library can be reconstructed to generate a 3D image in any point cloud processing software like Cloud Compare. The data, obtained in this manner, can be used to detect cracks and faults or other important features of the site which can be used for other applications.

7. ACKNOWLEDGEMENT.

We would sincerely like to thank the Director of Indian Institute of Remote Sensing, Dr. A Senthil Kumar, for giving us an opportunity to work in this organization and gain a significant amount of exposure to corporate work culture, ethics, and etiquettes to be followed while working in a professional environment.

We would also like to thank the coordinator of the our project at IIRS, Mrs. Shefali Agarwal, Head of the Department, Photogrammetry and Remote Sensing Division (PRSD), for guiding us through the program and supporting us throughout.

We would like to extend our deepest gratitude to our project in-charge, Dr. Raghavendra S, Scientist at PRSD, IIRS, for providing us with the opportunity to work with him on this project, and his guidance and mentorship during the same.

We are highly indebted to Dr. Praveen Goyal, the faculty in charge of the PS-1 program at IIRS, for his guidance during the project, and his helpfulness and responsiveness while addressing all the concerns we raised during the same. We also thank Mr. Siddhant Pundir, the co-instructor of the project for all the help and suggestions that he gave in favour of our project.

We would like to extend our gratitude to all other members of the organization, who have been co-operative with us during the program while having us as interns, and to the members of the Practice School Division, who have worked very hard for making this program a very fruitful one in terms of the experience gained by the students.

8. REFERENCES

8.1. Research papers -

[1]. Mobile mapping by Joshua I. France.

[2]. 6DOF Semi-Rigid SLAM for Mobile Scanning,Jan Elseberg, Dorit Borrmann, and Andreas N¨uchter

[3]. Registration of Long-Strip Terrestrial Laser Scanning Point Clouds Using RANSAC and Closed

Constraint Adjustment by Li Zheng , Manzhu Yu , Mengxiao Song , Anthony Stefanidis , Zheng Ji and Chaowei Yang

[4].Michael Bosse and Robert Zlot. Continuous 3D scan-matching with a spinning 2D laser scanner. [5]. In IEEE ICRA ’09, pages 4312 –4319, May 2009. B. J. Brown and S. Rusinkiewicz. Global Non-Rigid Alignment of3-D Scans. ACM Transactions on Graphics, 26(3):21, 2007.H. Chui and A. Rangarajan

[6]. A New Point Matching Algorithm forNon-Rigid Registration. CVIU, 89(2-3):114–141, 2003.B. Pitzer and C. Stiller.

[7]. Probabilistic mapping for mobile robots using spatial correlation models. In IEEE ICRA, pages 5402–5409, May2010.Todor Stoyanov and Achim J. Lilienthal. Maximum likelihood pointcloud acquisition from a mobile platform. In Proc. of the IEEE ICAR,

[8]. June 22–26 2009. J. A. Williams, M. Bennamoun, and S. Latham. Multiple View 3D Registration: A Review and a New Technique .In Proc. of the Int.Conf. On Systems, Man, and Cybernetics, 1999.

[9]. 6-DOF Localization for a Mobile Robot using Outdoor 3D Voxel Maps by Taro Suzuki, Mitsunori Kitamura, Yoshiharu Amano and Takumi Hashizume.

[10]. Processing the point cloud with Riscan Pro or Riprofile Cyber Mapping Lab UT-Dallas

[11]. A MAN-PORTABLE, IMU-FREE MOBILE MAPPING SYSTEM Andreas N¨uchter∗, Dorit Borrmann∗, Philipp Koch+, Markus K¨uhn+, Stefan May Informatics VII – Robotics and Telematics Julius-Maximilians University Wurzburg, Germany Faculty of Electrical Engineering, Precision Engineering, Information Technology Nuremberg Institute of Technology Georg Simon Ohm, Germany

[12]. Overcoming the Level Bubble: Terrestrial Laser Scanning Reference Frame Transformations Evon P. Silvia

8.2. Website Links -

http://www.lammertbies.nl/comm/info/GPS-time.html
http://www.ntp.org/
http://www.ntp.org/downloads.html
https://www.eecis.udel.edu/~mills/ntp/html/index.html
https://www.eecis.udel.edu/~mills/ntp/html/build.html
http://askubuntu.com/questions/25347/what-command-do-i-need-to-unzip-extract-a-tar-gz-file
http://askubuntu.com/questions/262068/how-to-extract-a-tar-gz-file
http://askubuntu.com/questions/499807/how-to-unzip-tgz-file-using-the-terminal
http://www.hostingadvice.com/how-to/untar-file-linuxubuntu/
http://www.cyberciti.biz/faq/linux-unix-bsd-extract-targz-file/
http://askubuntu.com/questions/974/how-can-i-install-software-or-packages-without-internet-offline
https://askubuntu.com/questions/86358/how-to-obtain-installed-package-files/86413#86413
http://askubuntu.com/questions/339/how-can-i-install-a-package-without-root-access
https://help.ubuntu.com/community/AptGet/Howto
https://help.ubuntu.com/community/InstallingSoftware
http://www.catb.org/gpsd/
http://www.catb.org/gpsd/installation.html

http://fossies.org/linux/gpsd/build.txt
http://www.catb.org/gpsd/troubleshooting.html
https://www.howtoinstall.co/en/ubuntu/trusty/gpsd-clients
http://askubuntu.com/questions/86822/how-can-i-copy-the-contents-of-a-folder-to-another-folder-in-a-different-directo
http://askubuntu.com/questions/80065/i-want-to-copy-a-directory-from-one-place-to-another-via-the-command-line
http://www.cyberciti.biz/faq/copy-folder-linux-command-line/
http://www.krizna.com/ubuntu/setup-dhcp-server-ubuntu-14-04/
https://help.ubuntu.com/community/isc-dhcp-server
https://help.ubuntu.com/community/dhcp3-server
http://askubuntu.com/questions/140126/how-do-i-install-and-configure-a-dhcp-server
https://www.chegg.com/tutors/opportunities/
http://askubuntu.com/questions/430853/how-do-i-find-my-internal-ip-address
http://www.howtogeek.com/howto/17012/how-to-find-your-ip-address-in-ubuntu/
http://askubuntu.com/questions/197628/how-do-i-find-my-network-ip-address-netmask-and-gateway-info
http://askubuntu.com/questions/278756/dhcp-server-not-starting
http://askubuntu.com/questions/398442/problem-configuring-dhcp-server-job-failed-to-start
http://www.webopedia.com/TERM/S/subnet_mask.html
http://www.linuxfromscratch.org/blfs/view/7.8/basicnet/dhcp.html
https://wiki.debian.org/DHCP_Client
http://www.computerhope.com/unix/dhclient.htm
https://linuxconfig.org/what-is-dhcp-and-how-to-configure-dhcp-server-in-linux
https://help.ubuntu.com/lts/serverguide/dhcp.html#dhcp-configuration
http://askubuntu.com/questions/449876/dhcp-server-client?rq=1
https://www.freebsd.org/doc/handbook/network-dhcp.html
http://askubuntu.com/users/logout
http://askubuntu.com/questions/112885/dhcp-server-not-routing-to-connect-to-internet-for-clients?rq=1
http://stackexchange.com/
https://help.ubuntu.com/lts/serverguide/dhcp.html
https://calomel.org/dhclient.html
http://serverfault.com/questions/329596/how-to-override-the-ntp-information-sent-by-dhcp-in-debian
http://doc.ntp.org/3-5.93e/ntpq.html
https://rbgeek.wordpress.com/2012/04/30/time-synchronization-on-ubuntu-12-04lts-using-ntp/
https://www.eecis.udel.edu/~mills/ntp/html/ntpq.html
https://help.ubuntu.com/lts/serverguide/network-configuration.html
http://manpages.ubuntu.com/manpages/wily/man5/dhclient.conf.5.html
http://manpages.ubuntu.com/manpages/wily/man5/dhcp-options.5.html
http://askubuntu.com/questions/30569/how-to-use-gps-receiver-bu-353
http://www.catb.org/gpsd/gpsd-time-service-howto.html
http://ubuntuforums.org/showthread.php?t=966569
http://stackoverflow.com/questions/17374120/error-trying-to-compile-v8-on-osx-lion-scons-no-sconstruct-file-found
http://stackoverflow.com/questions/17182799/scons-no-sconstruct-file-found
http://www.catb.org/gpsd/gpsd-time-service-howto.html#RFC-2783
https://tools.ietf.org/html/rfc2783
https://www.kernel.org/doc/Documentation/pps/pps.txt
https://lists.nongnu.org/archive/html/gpsd-users/2015-10/msg00018.html
http://unix.stackexchange.com/questions/120992/why-does-cat-ttyusb0-not-produce-output

http://unix.stackexchange.com/questions/200302/reading-dev-ttyusb0
http://www.catb.org/gpsd/gpsctl.html
http://askubuntu.com/questions/453072/what-is-nss-myhostname-and-why-is-it-not-installable

9. APPENDIX

All code, instruction files have been uploaded on the GitHub Repository:
https://github.com/SAGNIKMJR/Time-Synchronisation-of-Mobile-MappingSystem.git