Journal of Research in Electrical Circuits and Systems
Volume 7, Issue 1, January-April 2022

Evaluation of the Performance of a Low-Cost GPS Time Server Based on NTP

Pranam Tayade[1], Monika Dhar[2]
Student
Department of Electrical Engineering
Savitribai Phule Pune University, Ganeshkhind, Pune-411007, Inida
Corresponding Author's E-mail: *pranamtryade6@gmail.com*

Abstract

Time synchronisation is essential to keep a correct clock. A low-cost GPS NTP server has been accomplished in this work utilising a cheap arduino, GPS receiver, and ethernet shield. The performance of a low-cost GPS NTP server was compared to that of a commercial GPS NTP server (TM1000A). The findings indicated that both time servers had a 100% synchronisation success rate, with an average clock offset of -8,69 ms for the low cost GPS NTP server and -10,1538 ms for the TM1000A. However, as compared to a low-cost GPS NTP server, the TM1000A has a superior clock offset deviation area. The TM1000A has a lower clock offset deviation area of -8 ms to -12 ms, but the low cost GPS NTP server has a greater clock offset deviation area of -20 ms to +10 ms. We provide affordable GPS NTP servers as an alternative GPS NTP server for time synchronisation on computer networks, with manufacturing costs of less than 29 USD.

Keywords: *GPS time server, , time synchronisation, : arduino, minimal cost*

INTRODUCTION

Time is critical in computer network-based systems [1]. The precise time reference is required for all activities or processes [2]. Time must be reliable and precise for distribution networks, transportation, financial and legal operations, and a variety of other applications requiring widely spread resources [3–7]. Although every computer has an internal clock, many of them are inaccurate and experience time shifts. Furthermore, these clocks are prone to mistakes and incorrect resets [8].

Each computer has a time difference as a result. The clock rate discrepancy may exceed 40 microseconds per second [9]. As a result, time synchronisation is essential for computers in a computer network to maintain a precise clock in a time-critical system.

GPS has been widely used in both scientific research and business activities, such as navigating for accurate traffic flow prediction, surveys for errors in road measurements, high altitude UAV tracking and surveillance, mobile operation, interfacing with GPRS as wireless data collection, and more [10]. GPS (Worldwide Location System) is a global satellite navigation system that performs three primary functions: positioning, time transmission, and speed measuring [11].

Time synchronisation with satellite navigation systems has shown to be an effective method [12]. Because GPS satellites have numerous atomic clocks, they may be utilised as an extremely precise clock source. When the GPS receiver latches on to signals from at least four GPS satellites, the GPS receiver's position may be calculated. With the GPS receiver's capacity to receive very precise data from GPS satellites, exact position and time information may be received from any location [13]. Every hour on the local computer may be synced to the GPS clock using the GPS system [14].

The Network Time Protocol (NTP) is a protocol that enables time data to be sent between a server and a client on a computer network. NTP is extensively used on the Internet to synchronise clocks on computers and, more recently, a broad range of mobile devices [15].

NTP is used to manage computer networks with a wide range of message delays. The precision of time supplied by NTP in LANs is sub millisecond, and tens of milliseconds in WANs [16].

An NTP implementation might function as a main, secondary, or client server. A main server is synced to a UTC-traceable reference clock (e.g., Galileo, GPS, etc). A client connects to one or more upstream servers to synchronise. The timing gap between server and client must be smaller than 128 ms to maintain synchronisation [17].

Most researchers' attention in time synchronisation research is on the use of hardware and software, offset calculation, compensation for clock drifts, and message exchange design to achieve time synchronisation [16], with less attention paid to the development of inexpensive GPS NTP servers and GPS NTP server performance evaluation. Several experiments have been carried out in order to create a low-cost GPS-based time synchronisation board.

Refan [8] presented a low-cost GPS-based computer network time synchronisation board that uses the NTP Protocol. Refan [18] presents time synchronisation boards that may synchronise computer network time using NTP and have redundant techniques incorporated in the board for increased dependability. The board was tested by connecting it via Ethernet wire to a PC running Windows XP. The board was also tested in a virtual network of three computers using the freeware "Sun virtual box," and the NTP server successfully synced them all [8, 18].

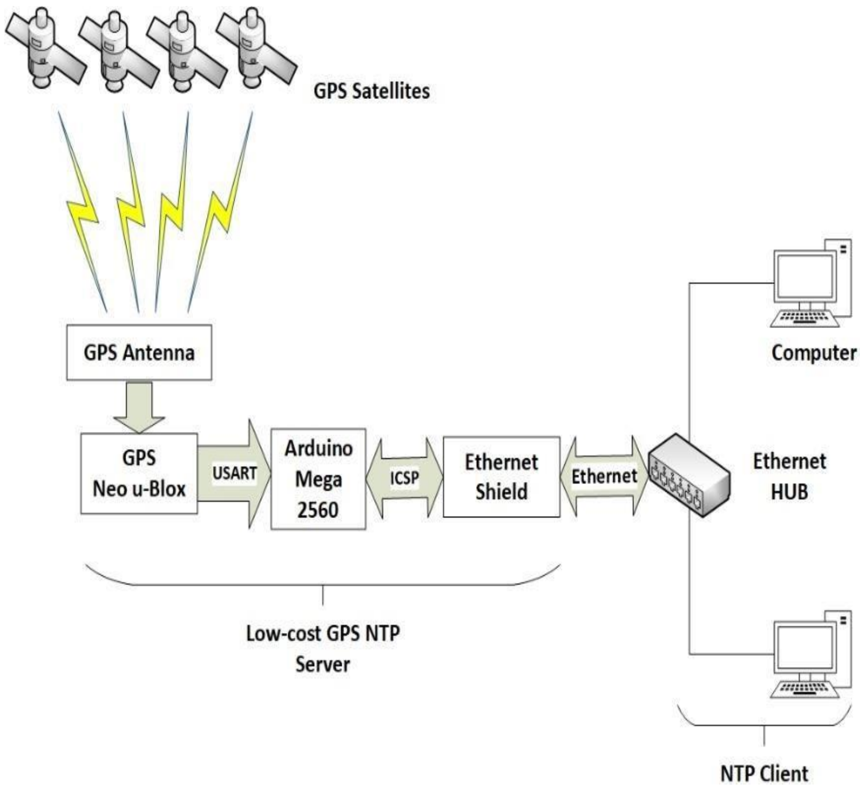
However, Refan's prior study [8, 18] did not report on the performance of low-cost GPS NTP servers vs commercial GPS NTP servers. The primary goal of this research is to create a low-cost GPS time server based on NTP as a stratum 1- NTP server for time synchronisation utilising an Arduino, an Ethernet shield [19], and a GPS receiver [20]. The performance of a low-cost GPS time server is also compared to that of a more costly GPS time server (TM1000A), with a cost difference of USD 29 vs USD 299.9.

Research Methodology

The research technique is divided into four phases. The first step is to create an architecture for a time synchronisation system. The architecture of GPS-based time synchronisation systems consists of two major components: the low-cost GPS NTP server and the NTP client. Figure 1 depicts the system's architecture.

GPS receivers receive time data from GPS satellites in the United States (u-Bblox GPS module Neo 6M). A GPS antenna is used by a GPS receiver to receive signals from GPS satellites.

The GPS receiver's time data was processed by the Arduino Mega 2560. If a client requests time data, the Arduino Mega 2560 will provide it to the client through Ethernet.



The second step involves creating a low-cost GPS NTP server using an Arduino, an Ethernet shield, and a u-Bblox GPS module Neo 6M. The third step involves utilising the NetTime programme to evaluate the low-cost GPS NTP server and commercial GPS NTP server (TM1000A) that are now available on the market. The fourth step involves determining the performance of a low-cost GPS NTP server and the TM1000A.

Flowchart for GPS NTP Server

Figure 2 depicts the flowchart of the main programme, which begins with the enablement of the Ethernet shield and the UDP function as a data communication channel. The GPS receiver validates the received data. If the GPS receiver receives data from the GPS satellites, the data is transformed into time data (year, month, day, hour, minute, and second), and if the time data is legitimate, the process of assembling and transmitting NTP messages to the client is initiated. If the data received by the GPS receiver is invalid, the GPS receiver's data validation procedure is restarted.

The flowchart of process NTP is shown in Figure 3, which starts with the verification of the NTP client request received through ethernet shield. If the NTP client's request does not exist, it will return to the main application. If an NTP client request is received, the process of reading the IP address and NTP port number is resumed. The next step is to fill up an NTP message with time data. The client that submitted the request is then delivered an NTP message through IP address and NTP port number. The procedure of verifying the data from the GPS receiver is then returned to the main application.

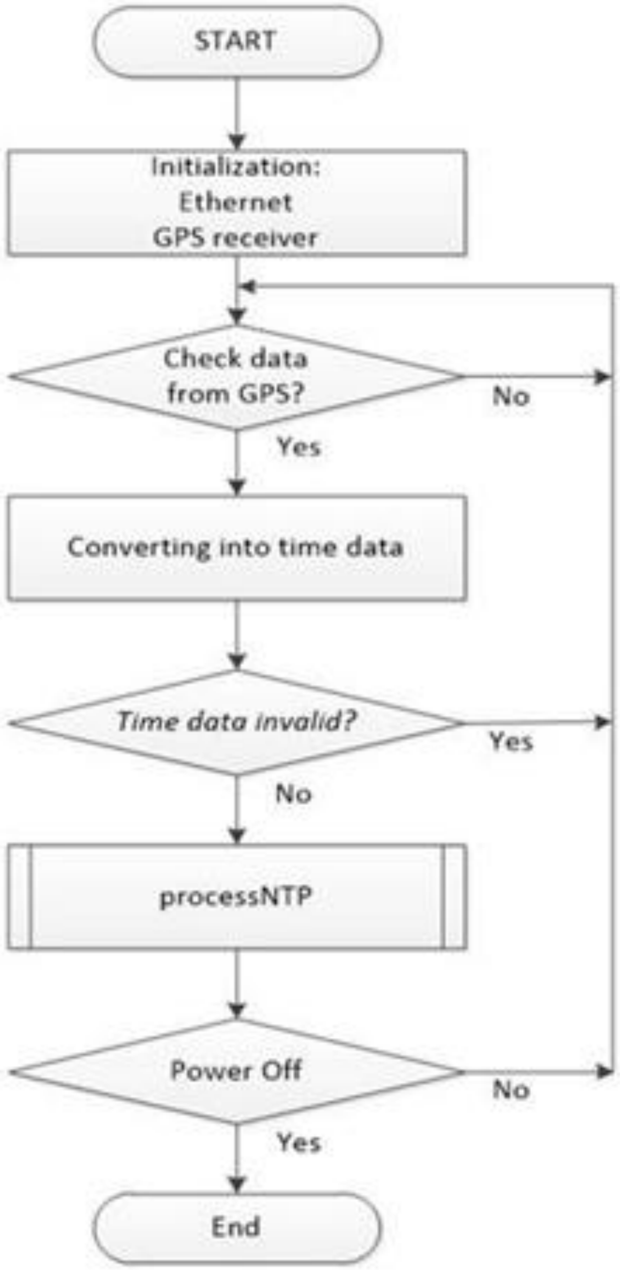


Figure 2. Flowchart of main program

Figure 3. Flowchart of process NTP

Constructing a Low-Cost GPS NTP Server

Arduino is an open source hardware and software platform that may be used to build a variety of gadgets. The arduino board has a microcontroller that can be programmed. Flowchart for GPS NTP Server

Figure 2 depicts the flowchart of the main programme, which begins with the enablement of the Ethernet shield and the UDP function as a data communication channel. The GPS receiver validates the received data. If the GPS receiver receives data from the GPS satellites, the data is transformed into time data (year, month, day, hour, minute, and second), and if the time data is legitimate, the process of assembling and transmitting NTP messages to the client is initiated. If the data received by the GPS receiver is invalid, the GPS receiver's data validation procedure is restarted.

The flowchart of process NTP is shown in Figure 3, which starts with the verification of the NTP client request received through ethernet shield. If the NTP client's request does not exist, it will return to the main application. If an NTP client request is received, the process of reading the IP address and NTP port number is resumed. The next step is to fill up an NTP message with time data. The client that submitted the request is then delivered an NTP message through IP address and NTP port number. The procedure of verifying the data from the GPS receiver is then returned to the main application.

Arduino programming language, which is comparable to C++ computer language [21], was used. Design of a low-cost GPS NTP server using an Arduino, an Ethernet shield, and the u-Bblox GPS module Neo 6M. ICSP (In-Circuit Serial Programming) or ISP is used to connect the Arduino to the Ethernet Shield (In-System Programming). While the Arduino and GPS receiver are connected through UART (Universal Asynchronous Receiver Transmitter). Table 1 depicts the link between the components. Figure 4 depicts the implementation of an NTP GPS Server based on an Arduino Mega 2560.

Table: - 1

Arduino Mega 2560 pin	Devices pin
Pin 18 Tx	Rx GPSreceiver module
Pin 19 Rx	Tx GPS receiver module
Pin 50 MOSI	D12 Ethernet Shield
Pin 51 MOSI	D11 Ethernet Shield
Pin 52 SCK	D13 Ethernet Shield
Pin 53 SS	D10 Ethernet Shield

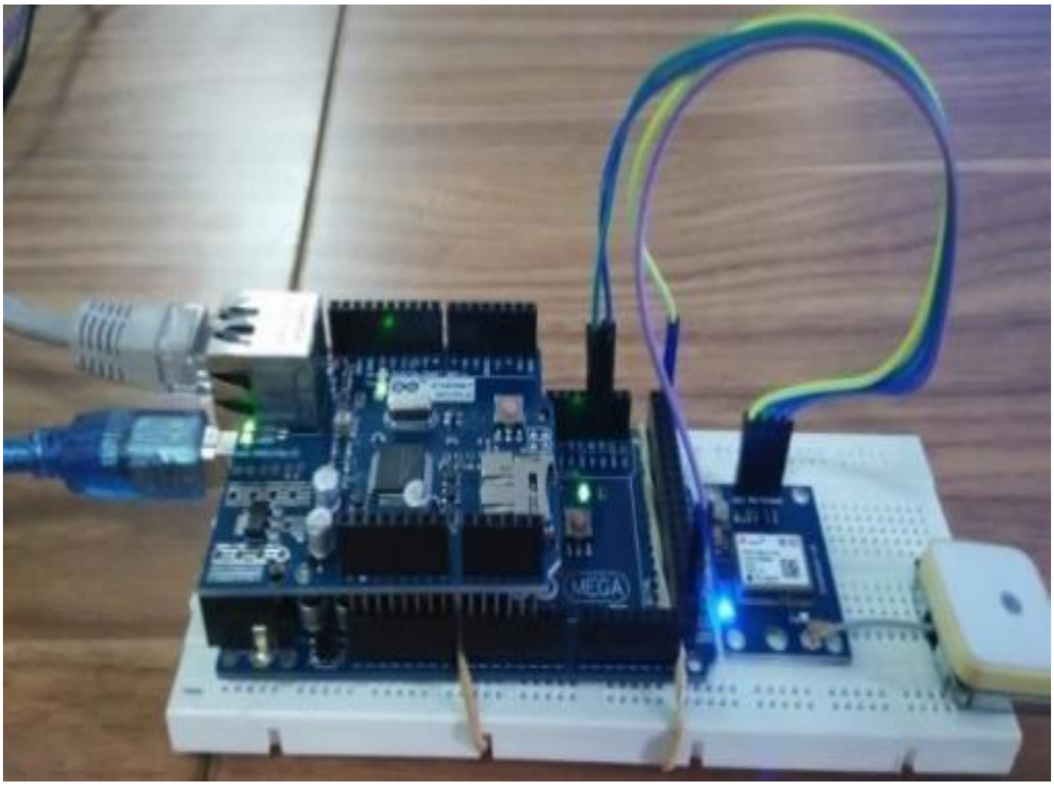


Figure 4. GPS NTP server based on arduino mega 2560

DISCUSSION AND RESULTS

Performance Analysis of a Low-Cost GPS NTP Server

In this research, a low-cost GPS NTP server was evaluated by connecting eight PCs as NTP clients to a low-cost GPS NTP server through computer network. The experiment was carried out in three parts. The first step involves eight computers watching the time when the GPS NTP server displays the time at 02:30:00 pm. Table 2 displays the timing difference for each customer. In the second step, eight computer units send a request to the GPS NTP server nearly simultaneously, when the GPS NTP server displays the time as 02:32:00 pm. The third step involves eight computers watching the time when the GPS NTP Server displays the time at 02:40:00 pm.

Table 3 displays the test results, which demonstrate that the low-cost GPS NTP server effectively synchronises all clients, which is consistent with prior findings [8, 18]. However, the GPS NTP server testing in this study is superior than the prior study since it was conducted on a computer network with eight genuine machines.

Table 2. Before Time Synchronization

Client	IP Address	Time
Computer 1	192.168.0.2	02:30:01 pm
Computer 2	192.168.0.3	02:29:46 pm
Computer 3	192.168.0.4	02:30:00 pm
Computer 4	192.168.0.5	02:30:05 pm
Computer 5	192.168.0.6	02:29:53 pm
Computer 6	192.168.0.7	02:30:02 pm
Computer 7	192.168.0.8	02:29:57 pm
Computer 8	192.168.0.9	02:29:54 pm

Table 3. After Time Synchronization

Client	IP Address	Time
Computer 1	192.168.0.2	02:40:00 pm
Computer 2	192.168.0.3	02:40:00 pm
Computer 3	192.168.0.4	02:40:00 pm
Computer 4	192.168.0.5	02:40:00 pm
Computer 5	192.168.0.6	02:40:00 pm
Computer 6	192.168.0.7	02:40:00 pm
Computer 7	192.168.0.8	02:40:00 pm
Computer 8	192.168.0.9	02:40:00 pm

Offset Clock

The clock offset is one of the variables retrieved from the client-server interaction for the estimate mechanism employed by NTP [22]. Figure 5 depicts the exchange of NTP time synchronisation packets between client and server. Clock offset is the time difference between the client's local time and the server time, which becomes changeable while performing time adjustments in the client. This fix will bring the two devices into sync. (1) [23] is used to calculate the clock offset.

$$Clock\ Offset = \frac{(T_2 - T_1) + (T_3 - T_4)}{2}$$	(1)

Defining T1 as the Client transmit time, T2 as the Server receive time, T3 as the Server transmit time, and T4 as the Client receive time. In this study, testing TM1000A and low-cost GPS NTP servers has been tested using NetTime application with interval pooling every 10 minutes.

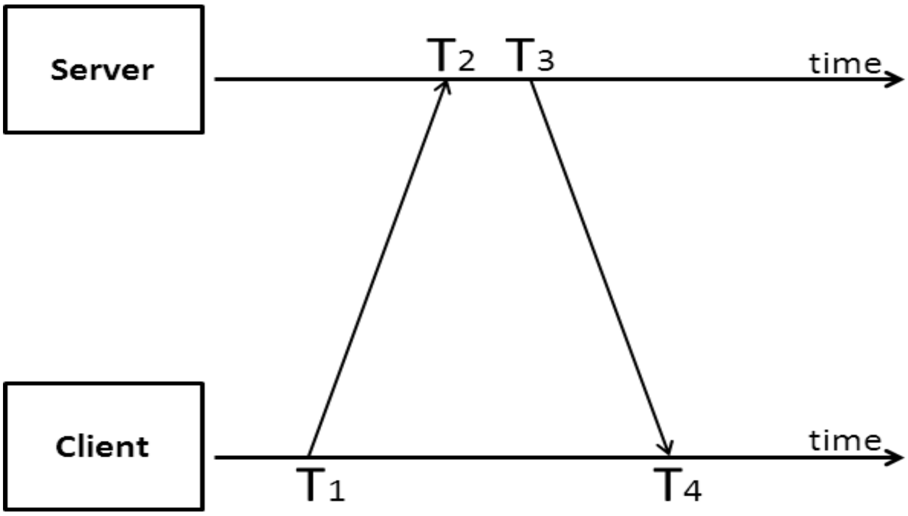


Figure 5: Client-server NTP time synchronisation packet exchange [22].

Testing of TM1000A

Figure 6 depicts the TM1000A GPS NTP server. The TM1000A was tested from September 20, 2016 at 03:14:49 pm to September 21, 2016 at 03:14:41 pm. As indicated in Figure 7, time synchronisation was only performed for one day since the findings demonstrate that the TM1000A was stable throughout time synchronisation with an average clock offset of -10,1538 ms. Figure 7 indicates that the synchronisation procedure may be completed successfully. Because there are no failures in the synchronisation process, the success rate is 100 percent. The majority of clock offset discrepancies are between -8 and -12 milliseconds. Only twice did it go outside of that zone, both times at the start of synchronisation.

Figure 6. TM1000A

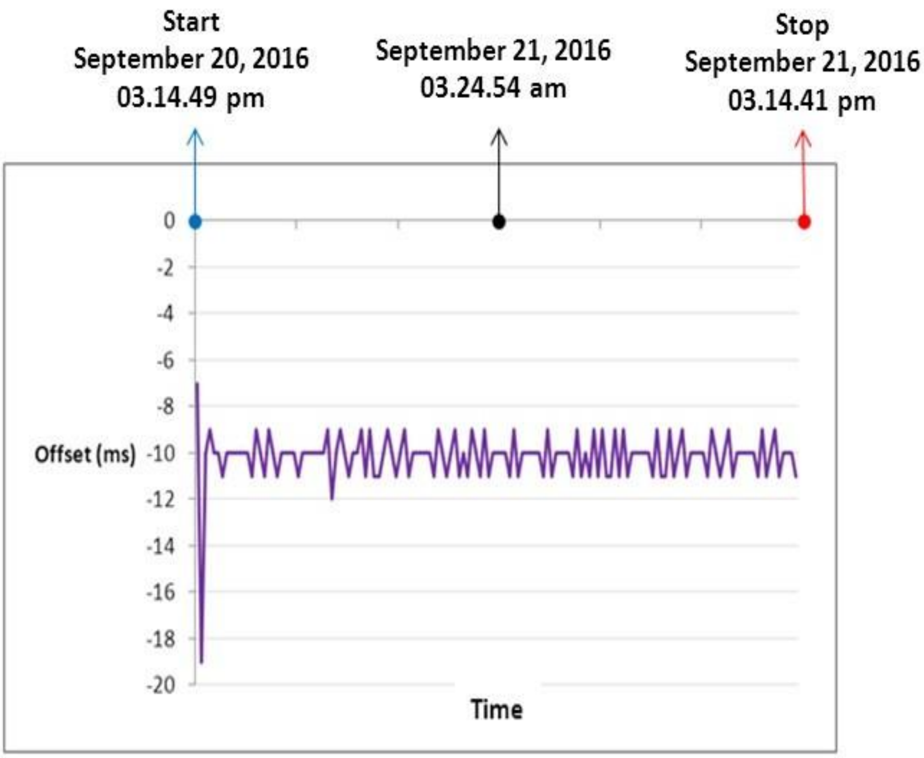


Figure 7. Testing the TM1000A

Testing of GPS NTP Server

Time synchronisation testing was completed in four days utilising a low-cost GPS NTP server and a PC, with interval pooling every 10 minutes and retry procedure in case of failure. The first day begins at 01:57:17 pm on Thursday, October 6, 2016 and ends at 01:48:26 pm on Friday, October 7, 2016. The second day begins at 01:57:17 pm on Friday, October 7, 2016 and ends at 01:53:02 pm on Saturday, October 8, 2016. Saturday, October 8, 2016 from 02.03.11 pm to Sunday, October 9, 2016 at 01.48.26 pm is the third day. The fourth day begins at 02.04.29 pm on Sunday, October 9, 2016 and ends at 01.55.38 pm on Monday, October 10, 2016.

Because there are no failures in the synchronisation process, the success rate of clock synchronisation is 100 percent, as illustrated in Figures 8, 9, 10, and 11. Over time synchronisation, the low-cost GPS NTP server has been fairly consistent. The majority of clock offset discrepancies fall between -20 and +10 milliseconds. Figure 8 shows the average clock offset of -9,4 ms on the first day. Only three times did offset occur beyond the range of -20 ms to +10 ms, twice before the start of synchronisation and once during synchronisation. Figure 9 shows the average clock offset of -8,83 ms on the second day. Figure 10 depicts the average clock offset of -8,03 ms on the third day. Only three occasions did the offset occur beyond the range of -20 to +10 ms during synchronisation. Figure 11 shows the average clock offset of -8,47 ms on the first day. Only one offset occurred beyond the range of -20 to +10 ms during synchronisation.

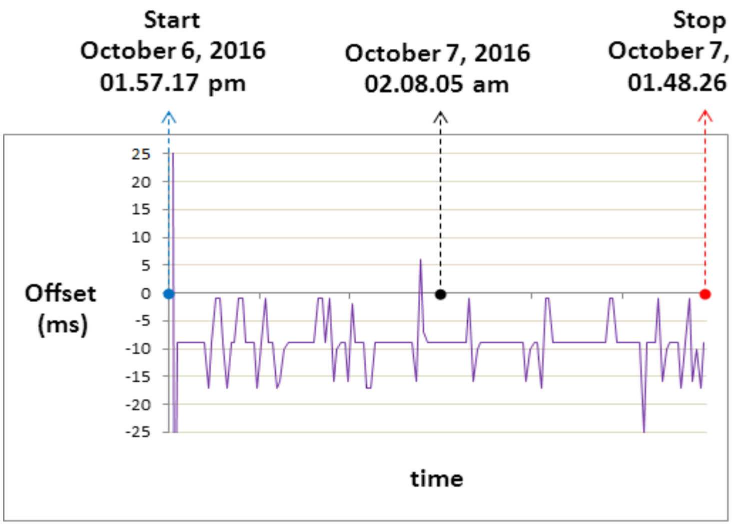


Figure 8. The first day of low cost GPS NTP server testing

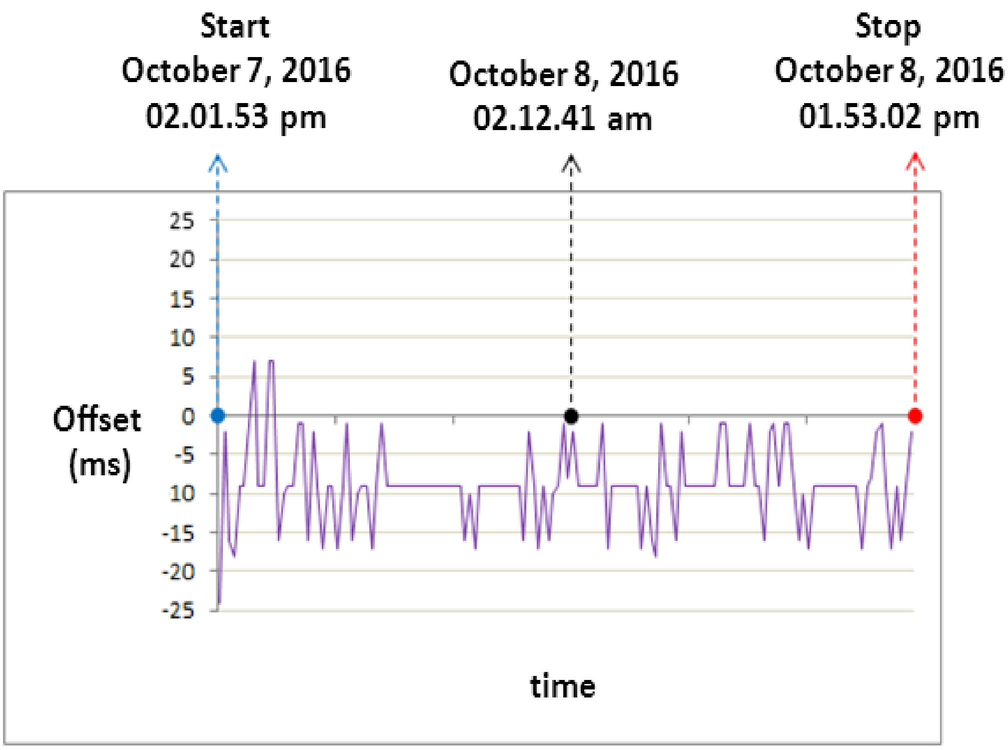


Figure 9. The second day of low cost GPS NTP server testing

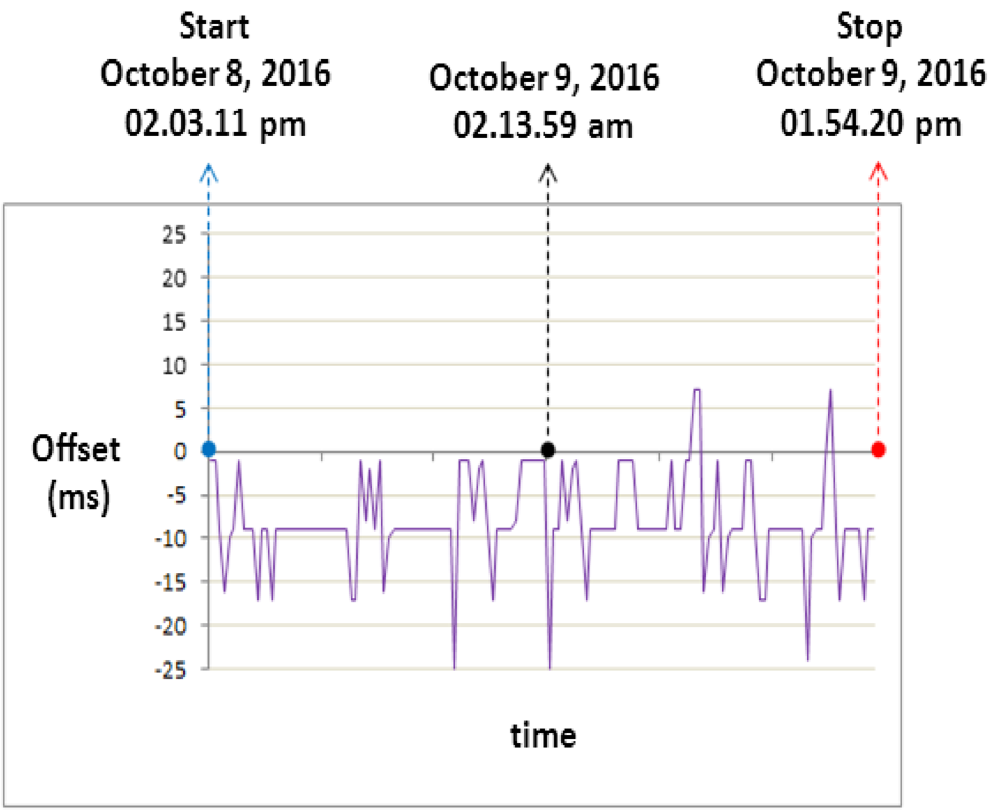


Figure 10. The third day of low cost GPS NTP server testing

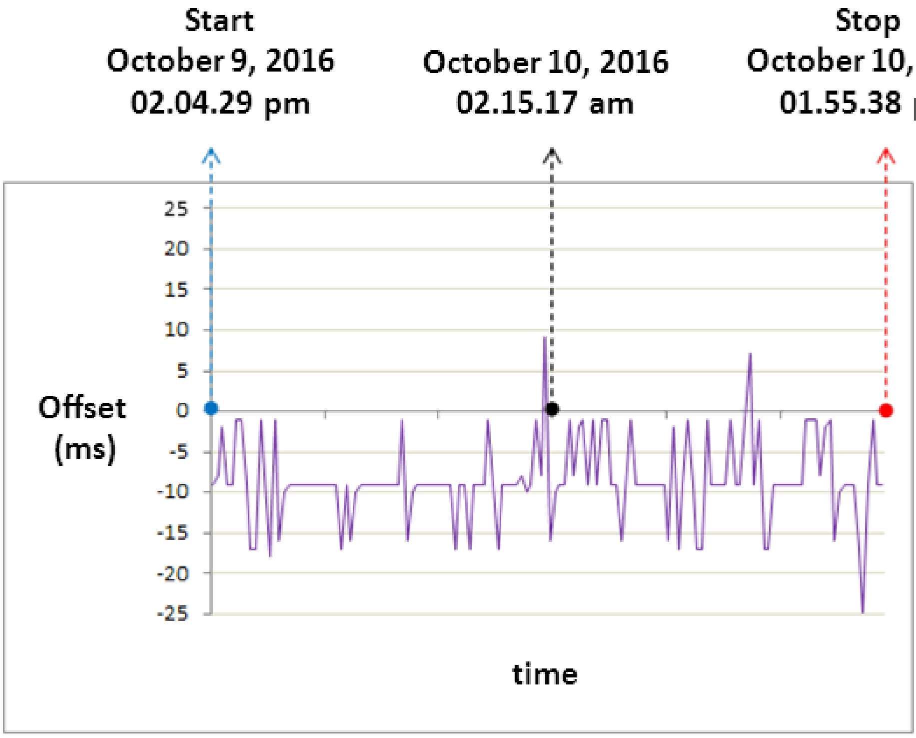


Figure 11. The fourth day GPS NTP server testing

CONCLUSION AND NEXT STEPS

Using a low-cost arduino, ethernet shield, and GPS receiver module, we successfully created a low-cost GPS NTP server for time synchronisation. We also evaluated the performance of low-cost GPS NTP servers to that of the TM1000A. Both GPS NTP servers are capable of syncing time with a success rate of 100%. The clock offset deviation area of the TM1000A, on the other hand, is better than that of a low-cost GPS NTP server since the TM1000A has a lower clock offset deviation area. This study will be expanded in the future to minimise clock offset deviation area on low-cost GPS NTP servers.

REFERENCES

1. Zhang X, Tang X, Chen J. Time synchronization of hierarchical real-time networked CNC system based on ethernet / internet. The International Journal of Advanced Manufacturing Technology. 2008; 36(11–12): 1145–1156.
2. Son S, Kim N, Lee B, Cho CH, Chong JW. A Time Synchronization Technique for CoAP-based Home Automation Systems. IEEE Transactions on Consumer Electronics. 2016; 62(1):10–16.

3. Veitch D, Ridoux J, Korada SB. Robust Synchronization of Absolute and Difference Clocks Over Networks. IEEE/ACM Transactions on Networking. 2009; 17(2): 417–430.

4. Helsby N, Dean W. Portable Instrumentation for Time Source Verification and Analysis. IEEE International Frequency Control Symposium Joint with the 21st European Frequency and Time Forum. Geneva: IEEE. 2007: 854–857.

5. Kusriyanto M. Smart Home Using Local Area Network (LAN) Based Arduino Mega 2560. 2nd International Conference on Wireless and Telematics (ICWT). Yogyakarta. 2016: 127–131.

6. Zuohu L, Jinming H, Jianwen L. High Precision Clock Synchronization and Control Based on GPS. International Conference on Electrical and Control Engineering. Wuhan. 2010: 1125–1128.

7. Lee J, Jeong Y, Nam K. Time synchronization method of Network Testing system by Standard Wave. 16th International Conference on Advanced Communication Technology. Pyeongchang: IEEE. 2014: 1136– 1139.

8. Wu J, Hu Y, He Z. The study of GPS Time Transfer Based on Extended Kalman Filter. Joint European Frequency and Time Forum & International Frequency Control Symposium (EFTF/IFC). Prague. 2013: 819–822.

9. Vito L De, Rapuano S, Tomaciello L. One-Way Delay Measurement : State of the Art. IEEE Transactions on Instrumentation and Measurement. 2008; 57(12): 2742– 2750.

10. Refan MH, Valizadeh H. Redundant GPS Time Synchronization Boards for Computer Networks. 19th Telecommunications Forum (TELFOR). Belgrade: IEEE. 2011: 904–907.

11. Odolinski R. Low-cost , high- precision , single-frequency GPS

BDS RTK positioning. GPS Solutions. 2017; 21(3): 1315–1330.

12. Generiwal S, Kumar R, Srivastava MB. Timing-sync Protocol for Sensor Networks. SenSys ’03 Proceedings of the 1st international conference on Embedded networked sensor systems. Los Angeles. 2003: 138–149.

13. Refan MH, Valizadeh H. Design and Implementation of a GPS Based DCS Network Time Synchronization Board. The 3rd Conference on Thermal Power Plants. Tehran. 2011: 1–6.

14. Son S, Kim N, Lee B, Cho CH, Chong JW. A Time Synchronization Technique for CoAP-based Home Automation Systems. IEEE Transactions on Consumer Electronics. 2016; 62(1): 10–16.

