GNSS Aided Precision Multi-clock Synchronization

Lihua Yang
School of Software
Northwestern Polytechnical University
Xian, China
mrpjf@mail.nwpu.edu.cn

Bing Li
School of Software
Northwestern Polytechnical University
Xian, China
libing@nwpu.edu.cn

Yue Yin
School of Microelectronics
Northwestern Polytechnical University
Xian, China
yinyue@nwpu.edu.cn

2021 4th International Conference on Hot Information-Centric Networking (HotICN) | 978-1-6654-0549-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/HOTICN53262.2021.9680871

***Abstract*—Precision Time Protocol (PTP) defined by the IEEE standard is a precision clock synchronization protocol (IEEE1588) for achieving high-precision synchronization in distributed systems by means of network communication and local computing technology. Although PTP system offers sub-microsecond clock synchronization in local area network by software-only implementation, it is difficult to meet the requirement of communication network which requires higher clock synchronization performance. This paper proposes a PTP system aided by Global Positioning System (GPS)/BeiDou Navigation Satellite System (BDS), which attains nanosecond performance.**

Keywords—clock synchronization, precision time protocol, IEEE 1588, distributed system

I. Introduction

With rapidly increasing scale of communication infrastructure, intelligence and automation emerges as the new form of distributed systems [1]. Hence, high precision clock synchronization become a crucial technique to offer lower delay, higher bandwidth and higher reliability [2]. Precision Time Protocol (PTP) is such a time synchronization technique enabling precision synchronization of distributed nodes accurately [3]. Compared with other time synchronization technologies such as NTP (Network Time Protocol) [4] and Global Positioning System (GPS) [5], PTP is characterized by easier configuration, faster convergence, higher accuracy and less consumption of network bandwidth and resources [6].

PTP is usually implemented in two ways: software-only implementation and hardware-assisted implementation [7]. A software-only implementation, usually developed by open-source software such as Linux PTP [8] and PTPd [9][10], has low implementation complexity. But the synchronization precision can only reach a precision on the order of sub-microseconds. On the contrary, a hardware-assisted implementation can provide more accurate timestamp, such as FPGA [11], physical layer chip [12] and integrated hardware platform [13][14]. In this way, the jitter and noise in the protocol stack can be reduced and the synchronization accuracy can be improved [15][16]. In summary, hardware-assisted implementation is able to achieve nanosecond-order precision.

This paper describes the PTP synchronization system aided by GPS/BDS timing module based on embedded Linux operating system to achieve the performance of nanosecond synchronization. The rest of this paper is organized as follows. Section 2 briefly discusses GPS/BDS timing technology and PTP clock adjustment mechanism and algorithm. Experimental setup, test method selected and experimental results are described in Section 3. Finally, section 4 provides a summary of this paper.

A. *Synchronization Technology Based on GPS/BDS*

After adjusted by the GPS/BDS multi-clock source timing module, the master clock can achieve more reliable and stable timing accuracy to synchronize the slave clock in the same LAN [17]. The combination of GPS/BDS synchronization technology and PTP can make up for the deficiency of both and give full play to their respective advantages. In other words, the physical distance of PTP system nodes can be expanded [18] in various scenarios.

B. *GPS/BDS Synchronization Technology*

GPS/BDS satellite timing module transforms satellite signals obtained from antenna into pulse per second (PPS) signals, and sends PPS to the clock nodes. The connection logic between timing module and core board is indicated in Fig. 1.

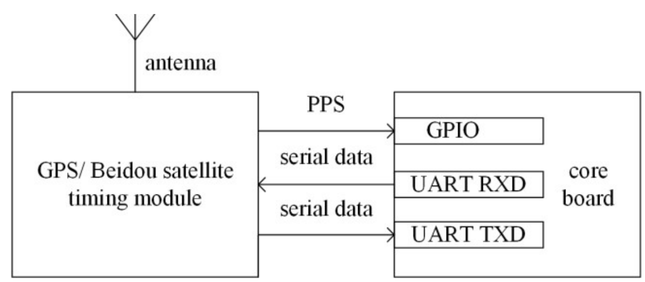

Fig. 1. Diagram of hardware structure

GPS/BDS satellite timing module receives satellite signals through antennas and then sends them to the core board in the form of PPS and serial data. The system adjusts the clock information according to the input signal which contains UTC time, longitude, latitude, altitude and other information. Then, the system uses PTP to send messages for synchronous communication to complete the multi-clock synchronization. The design principle of this paper is to minimize the use of modules to ensure higher accuracy and better reliability of the performance of the distributed system.

This paper is jointly supported by Youth Program of National Natural Science Foundation of China (Grant No. 61804126) and Aeronautical Science Foundation of China (Grant No.20200043053004, 20200043053005).

C. Synchronization Mechanism of Precision Time Protocol

A PTP clock can be either a master clock or a slave clock, depending on the state. By continuously exchanging clock synchronization messages between master and slaves, the local clock of PTP system is synchronized with the reference clock obtained from the network [19], as is illustrated in Fig. 2.

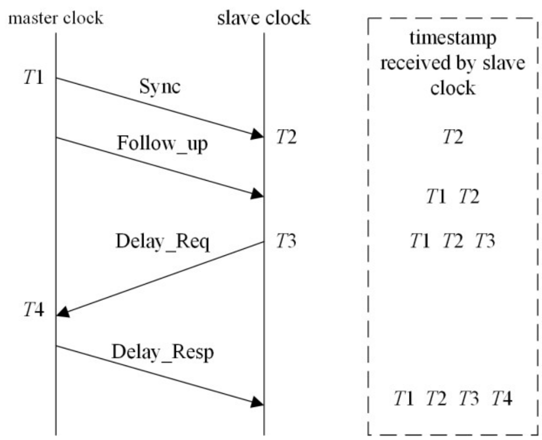

Fig. 2. Synchronization mechanism of PTP

The process of the PTP synchronization mechanism is as follows:

1) The slave listens to Sync message on the network. The master sends Sync messages at synchronization intervals and notes the timestamp T1 at which it was sent.

2) The slave receives the Sync message and notes the timestamp T2 of the receiving time.

3) The master sends the Follow_Up message carrying T1, from which the slave obtains the timestamp T1.

4) After receiving the Follow_Up message, the slave sends a Delay_Req message to the maste and notes the timestamp T3 of the sending time.

5) The master receives the Delay_Req message and notes the timestamp T4 of the receiving time.

6) The master sends a Delay_Resp message carrying T4 to the slave.

After receiving the Delay_Resp messages, the slave obtains $T1$, $T2$, $T3$ and $T4$ and computes the average *Delay* (one-way path) and the relative *Offset* between two clocks as follows:

$$T2 - T1 = Delay + Offset \tag{1}$$

$$T3 - T4 = Offset - Delay \tag{2}$$

$$Delay = (T2 - T1 + T4 - T3) / 2 \tag{3}$$

$$Offset = (T2 - T1 + T3 - T4) / 2 \tag{4}$$

The difference between the average *Delay* and the actual *Delay* is due to the asymmetry of the link, which leads to an error in the calculated value of the link delay. The error caused by the frequency of the oscillators is a small difference. For simplicity, we generally assume that the *Delay* from the master to salve is equal with the *Delay* from the slave to the master.

D. Best Master Clock Algorithm

The Best Master Clock Algorithm (BMCA) can determine whether a new external clock is better than the local clock according to the information embedded in the Announce message received by the local clock port [20]. This allows the local clock to compute the next state of its own port. The BMCA consists of two algorithms, data set comparison algorithm and state decision algorithm.

The data set comparison algorithm makes pairwise comparison based on the information embedded in the Announce messages, and compares the performance of the two clock data sets to determine the best data set [21].

Taking data set A and data set B as an example, the first step is to determine whether the clocks represented by the two data sets come from the same grand master (GM) clock, which is the highest-ranking master clock of all clocks in a domain with the highest synchronization accuracy and the best stability [22]. Choosing which obtains time from the GM is much more significant than finding which is the best clock to stabilize the algorithm. When synchronize with GM clock, these clocks store the configuration information of GM clock in its parent data set and transmitted to all clocks at the next level through Announce message [23]. The data set comparison algorithm running locally on all ports of clocks except transparent clock can compare the external clock data set contained in Announce message and the local data set to determine which of two data set is better rely on GM Priority1, GM identity, GM class, GM accuracy, GM offsetScaledLogVariance, GM Priority2, steps removed, identity of senders, identity of receiver, port number of receivers and other fields in data set.

D_0 is the local data set of clock node C_0, the best data set E_{rbest} for each port "r" of clock C_0 can be determined by data set comparison algorithm, as for the N port sets of C_0, the data set comparison algorithm can be used to determine the best data set E_{best} [24] from the N E_{rbest} data sets. When C_0 receives a Announce message, it shall first identify whether all ports have computed E_{rbest} since last such event. After that, C_0 computes E_{best} and is ready to apply the state decision algorithm. The logic of state decision algorithm is shown in Fig. 3.

The state of each port is dynamically updated based on the output provided by data set comparison algorithm [25]. When the state decision code is M1 or M2, it means that port r is in Master state and is the GM port of the system. At this time, the best data set is D_0. When the state decision code is M3, it means that port r is in Master state, but this port is not the GM port of the system. In this case, the best data set is E_{best}. When the state decision code is S1, port r is in Slave state. When the state decision code is P1 or P2, port r is in Passive state, indicating that there is a loop in the transmission path. In this case, the best data set is E_{rbest} [26]. After reaching a decision through the state decision algorithm, the clock C_0 shall update data sets for all ports and make required state changes in all ports and make required state changes in all ports according to the decision code. Therefore, the state decision codes provided by state decision algorithm and any data required for updating data sets are both the outputs of BMCA.

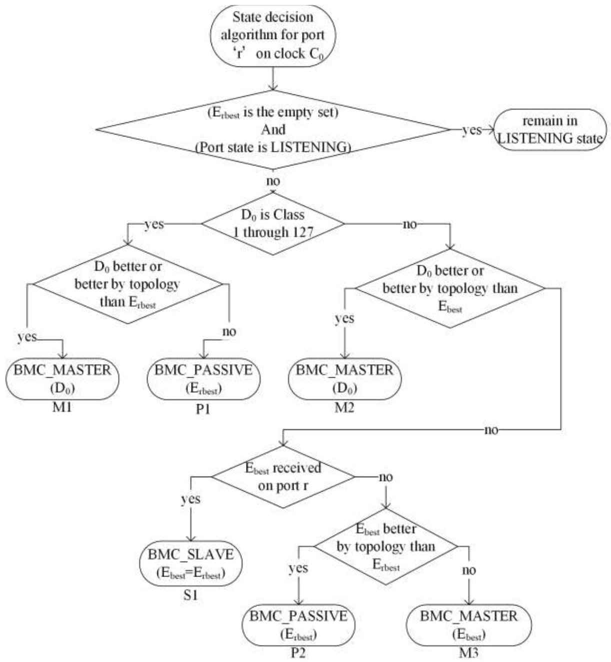

Fig. 3. State decision algorithm

II. IMPLEMENTATION OF PTP

In this section, we discuss the design and implementation of software and hardware. With the aid of hardware timestamp, we compare GNSS-PTP to standard PTP and evaluate the reliability of the scheme by experiments.

A. Experiment Design

1) Hardware

The hardware devices used in this paper are two development boards based on NXP i.MX6ULL controller and a desktop PC as a clock synchronization detection device. Both development boards can either be master clock or slave clock, and can complete time synchronization by switching messages carrying time information over Ethernet. The development board serving as the master clock is connected with the Global Navigation Satellite System (GNSS) module to obtain time signals of GPS/BDS satellite. The slave clock on another development board synchronizes with the master clock by using Linux PTP software, so as to achieve the synchronization between master clock and slave clocks.

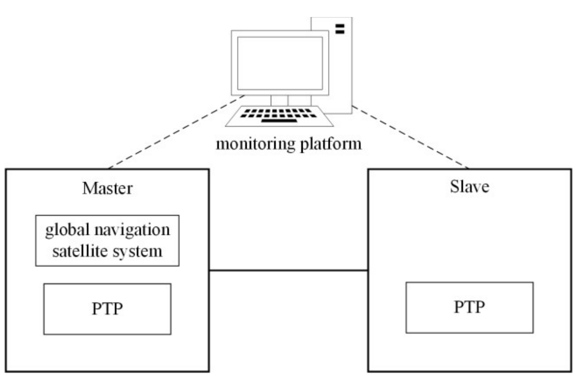

Fig. 4. Composition diagram of experimental platform

The two boards obtain hardware timestamps with nanosecond accuracy at the MAC layer. Each board has two RJ45 ports: eth0 runs the PTP protocol in direct connection mode, and eth1 connects to a switch or router that can keep the boards and computer on the same network. Finally, the synchronization results between master and slave clocks are recorded by the synchronization deviation monitoring platform. The specific environment is shown in Fig. 4.

2) Software

Linux PTP is an implementation of PTP designed for Linux-platform IEEE standard 1588. Linux PTP supports UDP/IPv4, UDP/IPv6, and raw Ethernet (Layer 2) transmission. In this paper, we use this stack to implement the synchronization processes, including delay request-response mechanism and BCMA. Considering the configuration of boards we used, we set up the protocol to transport over Ethernet and adopted the end-to-end delay measurement mechanism between the slave clock and the master clock for path delay correction. The synchronization interval is set to 1 second and other configurations are set by default.

B. Test Methods

After the Linux PTP runs stably, we continuously collected the time offset values of 4200 samples. These values are approximately Gaussian distributed and thus can be evaluated in terms of mean and variance.

First, the master clock and slave clock run Linux PTP in the absence of GNSS program running, and the probability statistics of offset between the master clock and the slave clock is depicted in Fig. 5. As shown in the figure, the values of offset is mainly concentrated within [-60ns, 60ns].

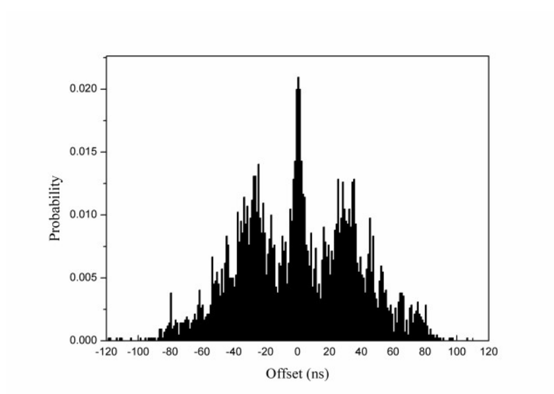

Fig. 5. Statistical histogram of distribution

When the GNSS signal is not accessible, the statistical analysis shows that the mean value of offset is -0.085ns and the variance of offset is 1346ns. The probability distribution of offset is illustrated in Fig. 6.

After the GNSS signal is obtained, the statistics histogram of offset is indicated in Fig. 7. As can be seen, the values of off- set is mainly in the range of [-40ns, 40ns], which demonstrates the stability of PTP system aided by external clock source. Further, the probability distribution of offset is illustrated in Fig. 8. The average value of offset is -0.176ns and the variance of offset is 1175ns.

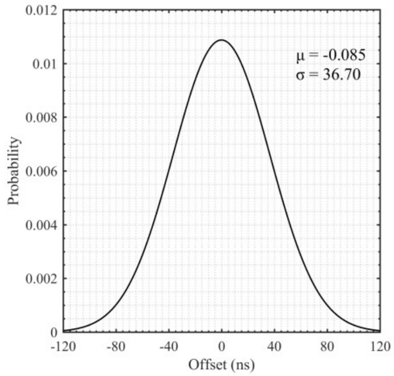

Fig. 6. Probability distribution of offset

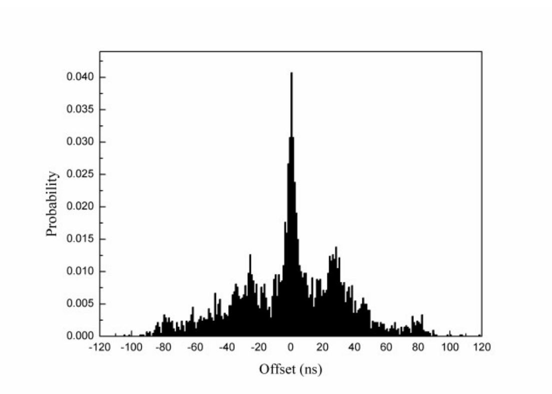

Fig. 7. Statistical histogram of distribution after running GNSS

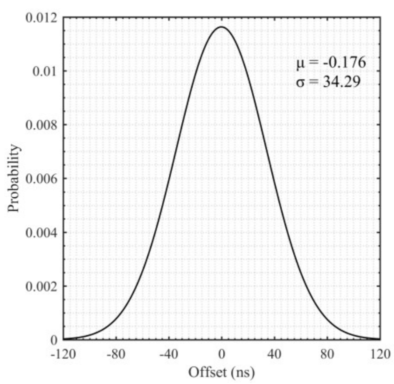

Fig. 8. Probability distribution of offset after running GNSS

C. Discussion

The stability of the clock is related to the external environment and its crystal oscillator. The stability of the reference clock in the PTP system directly affects the synchronization accuracy of all clocks. A GNSS-PTP method is proposed to reduce the time offset since it is a more stable subject to use atomic clocks with high performance. In these comparisons, we used the mean and variance σ^2 as performance indicators to evaluate time synchronization accuracy. In Table I, the mean and variance of time offset are -0.176ns and 1175ns, most offset values are within [−40, +40], indicating that GNSS-PTP enhances the noise reduction effect compared with standard PTP.

TABLE I. SUMMARY PERFORMANCE

	Min. (ns)	*Max. (ns)*	*Mean (ns)*	*σ^2 (ns)*
Non-GNSS	-119	106	-0.085	1346
GNSS	-105	118	-0.176	1175

III. CONCLUSION

In this paper, a master-slave clock model based on GPS/BDS satellite timing technology is established. The development board based on NXP i.MX6ULL controller and GPS/BDS satellite module is used as the hardware-assisted devices to achieve more stable and reliable accuracy. We evaluate the proposed approach in terms of offset distribution and clock stability. Based on the analysis of the clock synchronization offset curves and nonlinear fitting curves of offset, it can be seen that under the same experimental conditions, the clock model with more reliable external clock source outperforms the clock model without external clock source.

REFERENCES

[1] F. Girela-López, J. López-Jiménez, M. Jiménez-López, R. Rodríguez, E. Ros and J. Díaz, "IEEE 1588 High Accuracy Default Profile: Applications and Challenges," in IEEE Access, vol. 8, pp. 45211-45220, 2020.

[2] H. Puttnies, P. Danielis and D. Timmermann, "PTP-LP: Using Linear Programming to Increase the Delay Robustness of IEEE 1588 PTP," 2018 IEEE Global Communications Conference, 2018, pp. 1-7.

[3] K. O'Donoghue, D. Sibold and S. Fries, "New security mechanisms for network time synchronization protocols," 2017 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication, 2017, pp. 1-6.

[4] X. Jie, X. Liang, D. Lian and Z. Delin, "Research on network timing system based on NTP," 2017 13th IEEE International Conference on Electronic Measurement & Instruments, 2017, pp. 356-360.

[5] H. Guo and P. Crossley, "Design of a Time Synchronization System Based on GPS and IEEE 1588 for Transmission Substations," in IEEE Transactions on Power Delivery, vol. 32, no. 4, pp. 2091-2100, Aug. 2017.

[6] G. Zhaoqiang, "Wireless Network Time Synchronization Technology Based on IEEE1588V2," in Communications Technology, vol. 54, no. 1, pp. 147-151, 2021. (in Chinese)

[7] W. Xiantao, "Wireless Sensor Network Based on IEEE1588 Accurate Time Synchronization," in Modern transmission, vol. 00, no. 2, pp. 72-79, 2020. (in Chinese)

[8] L. Qing and R. Xinyang, "IEEE 1588 and a Dynamic Delay Correction Clock Synchronization Algorithm," 2019 IEEE 5th International Conference on Computer and Communications, 2019, pp. 442-446.

[9] R. Maegawa, D. Matsui, Y. Yamasaki and H. Ohsaki, "A Discrete Model of IEEE 1588-2008 Precision Time Protocol with Clock Servo using PI Controller," 2019 IEEE 43rd Annual Computer Software and Applications Conference, 2019, pp. 531-536.

[10] K. Lao and G. Yan, "Implementation and Analysis of IEEE 1588 PTP Daemon Based on Embedded System," 2020 39th Chinese Control Conference, 2020, pp. 4377-4382.

[11] H. Yin, P. Fu, J. Qiao and Y. Li, "The implementation of IEEE 1588 clock synchronization protocol based on FPGA," 2018 IEEE International Instrumentation and Measurement Technology Conference, 2018, pp. 1-6.

[12] X. Duan, Y. Guo, M. Liao, G. Lv, F. Li and J. Zou, "Research of precision time protocol in intelligent switchgears," 2017 4th International Conference on Electric Power Equipment - Switching Technology, 2017, pp. 890-893.

[13] P. Pandey, B. Pratap and R. S. Pandey, “Implementation of FreeRTOS based Precision Time Protocol (PTP) application as per IEEE1588v2 standards for Xilinx Zynq UltraScale Plus MPSoC devices,” 2019 International Conference on Communication and Electronics Systems, 2019, pp. 1968-1973.

[14] P. Pandey, B. Pratap and R. S. Pandey, “Analysis and Design of Precision Time Protocol System Based on IEEE1588 Standards,” 2019 International Conference on Communication and Electronics Systems, 2019, pp. 1963-1967.

[15] W. Xu, C. Ye, Z. Bin, “Software Implementation and Improvement of Clock Synchronization Based on IEEE 1588 Protocol,” in Techniques of Automation and Applications, vol. 39, no. 6, pp. 66-71, 2020. (in Chinese)

[16] M. Dong, Z. Qiu, W. Pan, C. Chen, J. Zhang and D. Zhang, “The Design and Implementation of IEEE 1588v2 Clock Synchronization System by Generating Hardware Timestamps in MAC Layer,” 2018 International Conference on Computer, Information and Telecommunication Systems, 2018, pp. 1-5.

[17] J. Fengnan, “Application of Beidou Clock Synchronization System in the Oilfield Thermoelectric Power Plant,” in Electrical Automation, vol. 38, no. 5, pp. 31-33, 2016. (in Chinese)

[18] L. Xujie, G. Baoshan, Y. Huaiqiao, Y. Hao, P. Shaojun and G. Yang, “Research and implementation of hybrid synchronization acquisition technology based on GPS/Bei Dou and PTP protocol,” in Proceedings of China High-tech Industrialization Association, Oct. 2015. (in Chinese)

[19] Z. Yi, “Based on IEEE1588 High Precision Time Synchronization Technical Analysis and Performance Testing,” in Journal of China Academy of Electronics and Information Technology, vol. 14, no. 6, pp. 605-610, 2019. (in Chinese)

[20] “IEEE standard for for a precision clock synchronization protocol for networked measurement and control systems,” IEEE Std 1588-2008, 2008.

[21] T. Kun, “Processor Research and design of IEEE1588 synchronous terminal system based on FPGA,” (Master Thesis) Hunan University, Changsha, Hunan, China, 2019.

[22] W. Jinqi, “Research on Time System of Distributed Testing System Based on IEEE1588v2 Protocol,” (Master Thesis) North University of China, Taiyuan, Shanxi, China, 2019.

[23] Z. Yan, “Industrial Ethernet Switch Precision Time Synchronization Technology and Its Implementation,” (Master Thesis) Shanghai Jiao Tong University, Shanghai, China, 2016.

[24] Z. Tuo, “Research on Slave Clock Synchronization Based on IEEE1588 Protocol,” (Master Thesis) Beijing University of Posts and Telecommunications, Beijing, China, 2018.

[25] W. Xu, “Research and Implementation of Clock Synchronization and Tigger for Distributed Test System,” (Master Thesis) North University of China, Taiyuan, Shanxi, China, 2020.

[26] K. Haiyang, “Design and practice of Internet Slave Clock on IEEE1588,” Dalian University of Technology University of Technology, Dalian, Liaoning, China, 2016.