

Letter

Distributed Nodes-Based Collaborative Sustaining of Precision Clock Synchronization upon Master Clock Failure in IEEE 1588 System

Kyou Jung Son ^{1,*}  and Tae Gyu Chang ² 

¹ Department of Electrical Engineering, Myongji University, Yongin 17058, Korea

² School of Electrical and Electronics Engineering, Chung-Ang University, Seoul 06974, Korea; tgchang@cau.ac.kr

* Correspondence: skj9865@naver.com; Tel.: +82-10-2503-9863

Received: 7 August 2020; Accepted: 12 October 2020; Published: 13 October 2020



Abstract: This paper proposes a distributed nodes-based clock synchronization method to sustain sub-microsecond precision synchronization of slave clocks upon master clock failure in IEEE 1588 PTP (precision time protocol) system. The sustaining is achieved by synchronizing the slave clocks to the estimated reference clock which is obtained from the analysis of distributed slave clocks. The proposed method consists of two clock correction functions (i.e., a self-correction and a collaborative correction, respectively). Upon master failure, the self-correction estimates a clock correction value based on the clock model which is constructed during normal PTP operation. The collaborative correction is performed in the preselected management node. The management node estimates a reference clock by collecting and analyzing clock information gathered from the other slave clocks. The performance of the proposed method is simulated by computer to show its usefulness. It is confirmed that the fifty (50) clock model-based collaborative correction maintains 10^{-6} second PTP accuracy for 10 min prolonged period after the master failure when tested with clock offset variations less than 50 ppm.

Keywords: precision time synchronization; distributed sensing environment; IEEE 1588 precision time protocol

1. Introduction

A precision time synchronization technique is considered essential in industrial applications where various distributed data are used. In a future LAN-based distributed sensing environment in which lots of sensors participate, preciseness of time synchronization is emphasized for stable and accurate operation of systems. In general, implementation of precision time synchronization in a network environment is considered difficult since many delay features such as propagation delay, processing delay, etc., cause time synchronization error in measurement data.

The IEEE 1588 PTP (precision time protocol) standard [1] has been published for precision time synchronization (PTS) in the LAN-based network. The IEEE 1588 PTP provides sub-microsecond synchronization accuracy supported from both hardware and software to overcome the delay problem in a network environment. As a master-slave relation-based time synchronizing method, the IEEE 1588 PTP maintains its preciseness by synchronizing slave clock's time and frequency to a master clock as shown by many previous researches [2–5]. Because of its preciseness, many time synchronization profiles based on the IEEE 1588 PTP, especially for power system applications, have been specified [6–8].

It is shown that the degraded time synchronizing performance directly leads to maloperation of industrial applications such as protection and control of power system apparatus [9–11]. Improved performance of the IEEE 1588 PTP is achieved in many ways, e.g., dealing with external disturbances

and attack [11–13], providing enhanced clock synchronization methods [14,15], reflecting asymmetric network delay [16,17], etc. Master failure is also an important issue for reliable provision of PTS. The majority of previous research related to master failure is confined to the subject of providing backup master and preventing deterioration of PTP synchronization performance [18,19]. The direct provision of a sustaining method upon total failure of master clock without the support from backup master is eventually needed to guarantee the reliable operation of LAN-based future distributed systems.

This paper proposes a distributed nodes-based collaborative PTS sustaining method. The proposed method is based on combined utilization of self-correction and collaborative correction. A prediction algorithm is adopted to estimate the clock drift model during the normal PTP operation period before master failure. Immediately after master failure, the self-correction prolongs PTS using the estimated clock model until a preselected management node performs collaborative correction.

Clock data of the distributed slaves are gathered for the collaborative correction. The collaborative correction estimates a reference clock by exploring the spatially distributed clock drift characteristics (i.e., offset, transient slope and additive noise of individual clock). The estimated reference clock enables the PTS sustaining via the application of the IEEE 1588 PTP-based synchronization.

The management node is selected as a master clock through the best master clock algorithm (BMCA) [1]. Slave clock information required for the collaborative correction in the management node can be obtained through TLVs (type, length, value) defined in the IEEE 1588 PTP [1]. The TLV packet propagation delay between the management node and slave clocks must be compensated. Compensation of the packet propagation delay can be achieved by using peer-to-peer delay mechanism of the IEEE 1588 PTP applying the pre-calculated link delay from adjacent nodes.

The performance of the proposed method was evaluated through computer simulations. It is confirmed that the 50 clock model-based collaborative correction maintains 10^{-6} second PTP accuracy for a 10 min prolonged period after master failure when tested with clock offset variations less than 50 ppm. The operability and the performance of the proposed method are also validated from an embedded DSP hardware operation test. The operation test showed that the proposed algorithm sustains sub-microsecond precision for 124 s after master failure. This is considered very long compared with 20 s which is measured when the proposed algorithm is not applied. In contrast to other redundant equipment-based backup solutions such as backup master, high-availability seamless redundancy (HSR) [20], parallel redundancy protocol (PRP) [20] etc., the proposed algorithm is considered a software-based solution which can provide PTS sustaining means upon the total failure of master clock and backup equipment. Therefore, the proposed PTS sustaining method is expected to provide a significant contribution to the reliable operation of industrial applications.

2. Distributed Nodes-Based Collaborative PTS Sustaining Method

In this section, basic principles of the proposed distributed nodes-based collaborative PTS sustaining method are described. The conceptual diagram of the proposed method is described in Figure 1. Temporal and spatial data of distributed nodes are used to compensate clock disturbances and eventually provide collaboration-based reference clock information. The application of linear prediction algorithms to the temporal data can provide estimation results of clock frequency, which is very useful to maintain precision synchronization during the clock self-correction period. Ensemble averaging is a good candidate for the effective exploration of the spatially distributed clock drift characteristics (i.e., offset, transient slope and the additive noise of individual clocks) providing accurate reference clock estimation by suppressing clock offsets and disturbances. The collection of the slave clocks' data can be achieved by using management functions defined in the IEEE 1588 PTP standard [1]. Multirate sampled data is used to compensate low-rate disturbances which are distinguished from clock's statistical features.

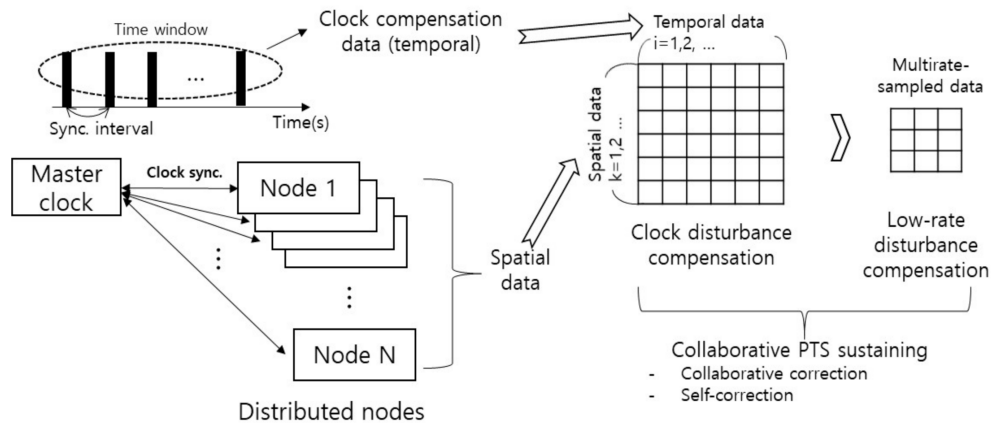


Figure 1. Conceptual diagram of the proposed precision time synchronization (PTS) method.

The operation process of the proposed method is illustrated in Figure 2. The proposed method is divided into two functions: the clock self-correction and collaborative clock correction. A prediction algorithm is adopted to estimate the clock drift model during the normal PTP operation period. Immediately after master failure, the self-correction prolongs PTS using the estimated clock model until a preselected management node performs collaborative correction. The collaborative clock correction algorithm starts after the activation of the management node. The management node is activated when it becomes a master clock by the result of the BMCA of the IEEE 1588 PTP [1]. The initialization process of the collaborative correction algorithm includes adaptive configuration of PTP parameters and sustaining PTP specifications including precision class and sustaining duration requirements. The management node collects the spatiotemporal data from distributed slave clocks to estimate the collaboration-based reference clock. After the estimation, the management node and slave clocks synchronize their clocks to the reference clock, thus PTS sustaining is achieved.

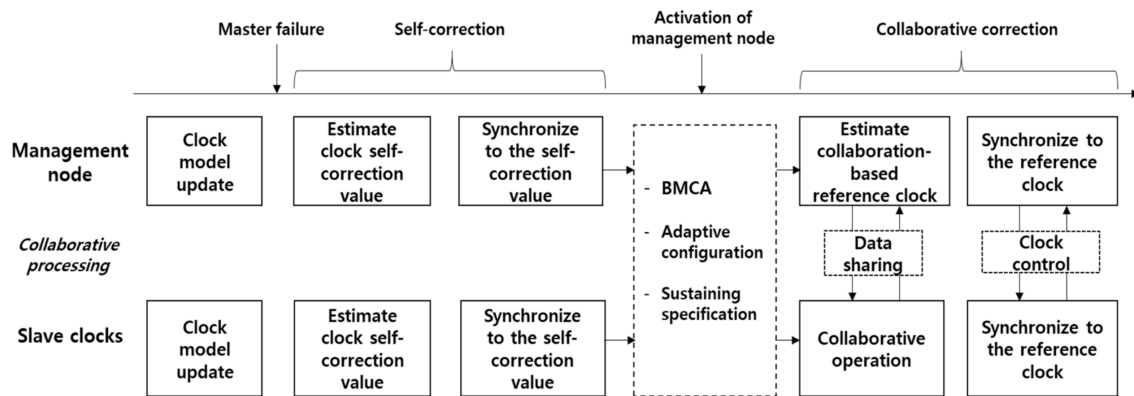


Figure 2. Operation process of the proposed method.

2.1. Clock Self-Correction

The proposed clock self-correction constructs a slave clock's own clock model by using clock compensation data obtained from the PTP-based synchronizing operation. As shown in Equation (1), the constructed clock model reflects external disturbances and clock's characteristics such as offset and drift. After master failure, the clock model is used to maintain PTS by estimating a clock self-correction value.

$$F_n(k) = F_{ref} + D_n(k) + v_n(k), \quad (1)$$

where k denotes the discrete time index which is updated with synchronization period T_{sync} . $F_n(k)$ denotes the frequency of n th slave clock. F_{ref} denotes the reference frequency. $D_n(k)$ denotes

the frequency deviation parameter which contains both offset and drift. $v_n(k)$ denotes the additive disturbance.

It can be assumed that clock frequency does not have much variations resulting in a quasi-stationary characteristic, which enables convergence of adaptive filtering algorithms under a controlled environment preventing external airflow and keeping minimized variations on temperature and humidity. Therefore, the frequency deviation parameter $D_n(k)$ can be estimated with its past information as shown in Equation (2). After master failure, the self-correction algorithm provides the estimated frequency deviation value, $\hat{D}_n(k)$, to compensate $D_n(k)$.

$$\hat{D}_n(k+1) = D_n(k)w_n^1(k) + D_n(k-1)w_n^2(k) \cdots + D_n(k-(M-2))w_n^{M-1}(k), \quad (2)$$

where $w_n^l(k)$ denotes l th prediction-filter coefficient of n th slave clock. M denotes the number of filter coefficients. The filter coefficients are updated during the normal PTP operation.

The flow chart of the self-correction is shown in Figure 3. During the periodical PTP-based synchronizing operation, a slave clock updates its clock model by using clock correction values obtained from the result of the synchronizing process with a master clock. The clock model update process is repeated until the occasion of master failure. After master failure, the clock self-correction is activated instead of the master-slave relation-based synchronizing process. The self-correction value is estimated from the constructed clock model. The slave clock synchronizes its clock parameter to the estimated self-correction value. The clock model must not be updated after the master failure since there is no reference information.

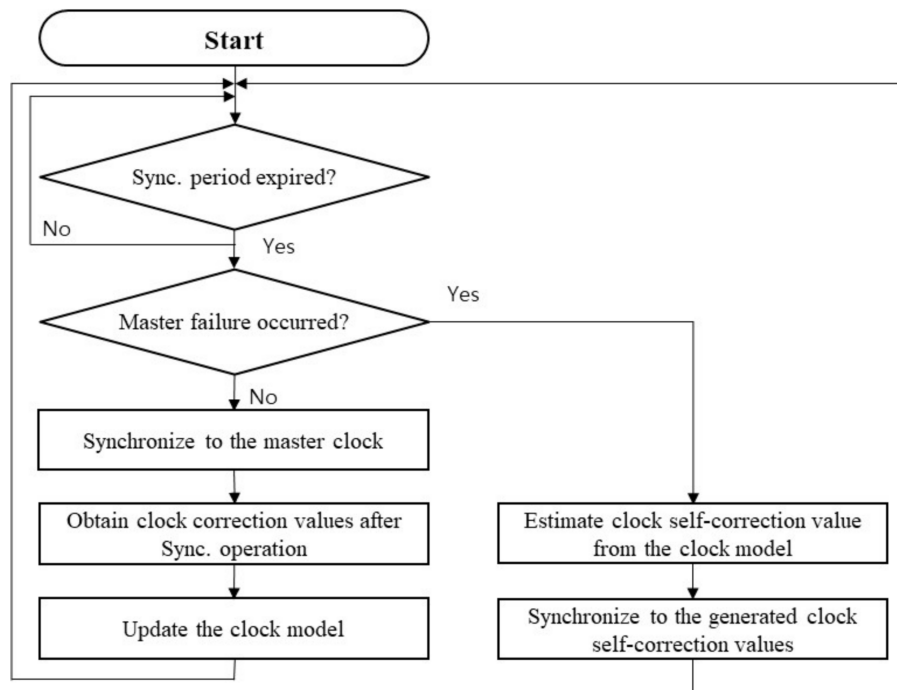


Figure 3. The flow chart of the proposed clock rate self-correction algorithm.

The performance of the clock self-correction mainly depends on the accuracy of the constructed clock model. The main target of the modeling is estimating oscillator's characteristics such as offset and drift. Unless the surrounding environment of PTP equipment is abruptly varying, the oscillator's characteristics can be estimated by application of adaptive modeling techniques to the clock correction values obtained from the PTP process. Figure 4 is an example of the clock modeling process. The adaptive filter update block adjusts the clock model as the new clock correction value is periodically calculated.

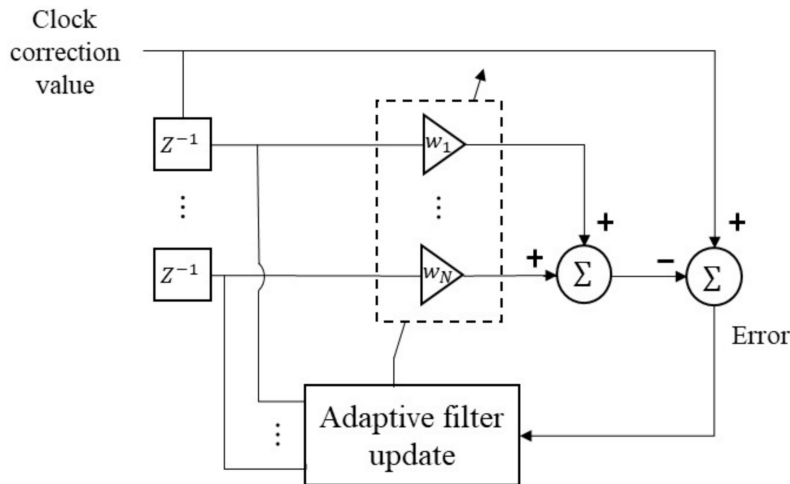


Figure 4. An example of the clock modeling process.

The LMS algorithm is considered a proper choice for the slowly changing clock model in view of convergence and accuracy performance as shown in Equation (3).

$$w_n(k+1) = w_n(k) + \mu \times D_n(k) \times J_n(k), \quad (3)$$

where μ denotes a step-size parameter. $w_n(k)$ denotes the prediction-filter coefficient vector of n th slave clock. $D_n(k)$ denotes the vector of the frequency deviation parameter of n th slave clock. $J_n(k)$ denotes the frequency prediction error of n th slave clock. In general, the application of other optimum algorithms will show comparable or better performance and benefit from the utilization of more data and stochastic information than the utilization of instantaneous data in the LMS algorithm. However, with the clock used in this research, it is shown that the clock frequency does not have much variations resulting in a quasi-stationary characteristic over several minutes by providing proper control of the clock operating environment.

2.2. Collaborative Clock Correction Algorithm

To maximize prolonged PTS, the proposed collaborative clock correction must be performed based on the results of the self-correction. Reference clock estimation by exploring the spatially distributed clock drift characteristics of an individual clock is a key element constructing the collaborative correction. Ensemble averaging of spatially distributed clock characteristics is adopted in the implementation of the collaborative correction to suppress the effect of clock offsets and disturbances as shown in Equations (4)–(6).

$$\text{Ensemble average of clocks} = \frac{1}{N} \sum_{n=1}^N t_n(k); \quad (4)$$

$$t_n(k) = t_n(k-1) + T_{sync} + E_n(k); \quad (5)$$

$$E_n(k) = \frac{F_n(k-1) - F_{ref}}{F_{ref}} \times T_{sync}, \quad (6)$$

where, $t_n(k)$ denotes the measured time at time index k of n th slave clock. T_{sync} denotes the PTP time synchronization period. $E_n(k)$ denotes the time error measured at time index k . Effectiveness of applying the ensemble averaging with Equations (4)–(6) is supported considering that the behaviors of widely spread clocks tend to show increased independence.

The flow chart of the proposed collaborative clock correction is shown in Figure 5. After master failure, the management node gathers the clock data of slave clocks through the data sharing process. TLV packets, which are used to get or set the PTP clock's parameters as defined in the IEEE 1588 PTP, are used to gather the clock information. Ensemble average of the gathered clock information provides

the estimated collaboration-based reference clock. The reference clock is used to synchronize all the slave clocks via the IEEE 1588 PTP-based synchronization.

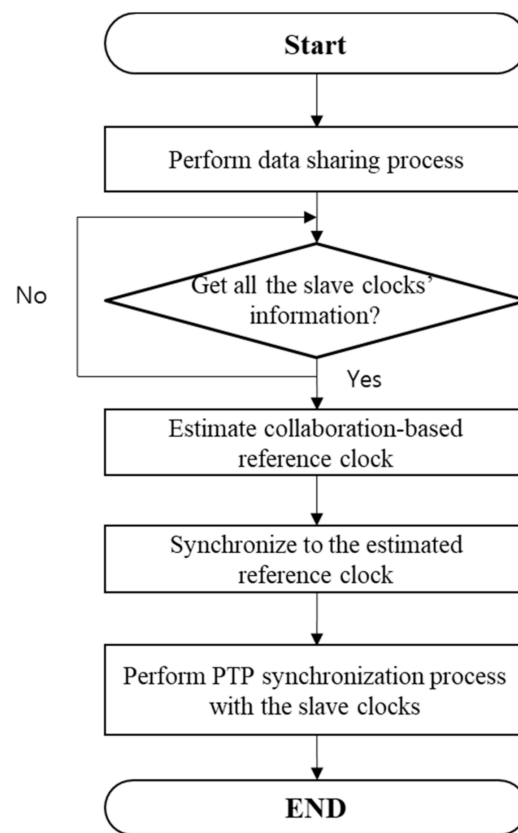


Figure 5. The flow chart of the proposed collaborative clock correction process.

Propagation delay occurring in the collection of the clock information must be compensated to enhance the preciseness of the generated reference time. Under a symmetric network condition, propagation delay compensation can be achieved by using peer-to-peer delay mechanism of the IEEE 1588 PTP. Every node can compensate propagation delay from adjacent nodes without requiring additional processes since they know link delay occurs against their adjacent nodes in the PTP environment where peer-to-peer delay mechanism is adopted. A priori characterization of end-to-end propagation delay, which requires additional execution of the end-to-end delay measurement process of the IEEE 1588 PTP, can also be utilized in the case of a heterogeneous PTP network environment.

The two functions of the proposed method (i.e., the clock self-correction and the collaborative clock correction) should be performed together to gain the maximized sustaining ability of PTS. It is especially important that the clock self-correction process be performed as soon as possible after the master failure. This is because sub-microsecond preciseness of clock is only maintained for a very short time without reference information.

3. Simulations

In this section, performance of the proposed PTS sustaining method was evaluated with computer simulations. The operation of the proposed method is also tested with an embedded DSP hardware system. To verify the excellence of the proposed method, time synchronization performance was compared with a PTP slave clock which only performs the original PTP process.

3.1. Performance Evaluation with Computer Simulations

Performances of the proposed PTS sustaining algorithm were evaluated with a Matlab program. The tab number and a step-size of the LMS algorithm was set to 8 and 0.005, respectively. The time synchronization period was set to 1 s. It is assumed that the propagation delay that occurred during the clock data sharing is compensated for.

The clock model used in the simulation was created by using a 2-tab IIR filter, where the filter input was a randomly chosen offset value to reflect a saturation characteristic. The nominal frequency of the clock was set to 20 MHz. A zero-mean white gaussian noise with a standard deviation of 1 Hz was added as a disturbance. Two parameters of a clock servo algorithm (i.e., a proportional and an integral constant) were set to 0.1 and 0.3, respectively. Examples of the clock models are shown in Figure 6.

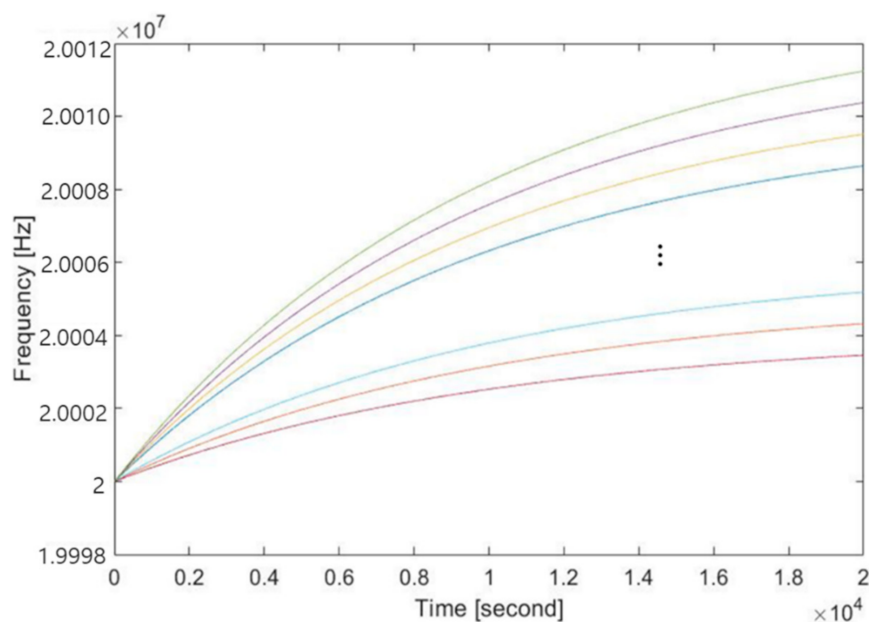


Figure 6. An example of the clock models used in the simulation.

Time synchronization performances of PTP clocks were observed during the simulation of which the duration was 5000 s. The normal PTP operation was performed for the first half of the duration (i.e., 2500 s). At 2501 s, the self-correction algorithm was activated because of the master failure. The collaborative correction algorithm operated after 20 s of the self-correction algorithm.

The time synchronization performances of three PTP slave clocks (i.e., a management node with the proposed PTS sustaining algorithm, a self-correction only applied slave clock and a normal slave clock without the proposed method) were compared. Fifty clocks including the management node participated in the collaborative correction. All 50 clocks synchronized to the same master during the normal PTP operation. After the master failure, the management node became a new master and the remaining 49 clocks synchronized to the management node.

The results of the performance comparison are shown in Figure 7. Before the master failure, the PTP clocks maintained sub-microsecond synchronizing performance by the normal PTP operation. The effectiveness of the proposed PTS sustaining algorithm is well shown by comparing the slopes of the time synchronization error of the PTP clocks after the master failure. The PTP clock with the proposed algorithm has the lowest time synchronization error slope. On the other hand, the time synchronization error of the PTP clock without the proposed algorithm drastically increased so its sub-microsecond synchronization performance was sustained for a relatively short time. The time synchronization error of the self-correction only applied PTP clock increased as the model error

between the estimated clock and the real clock increased. The collaborative correction must substitute the self-correction algorithm before the model error results in large time error.

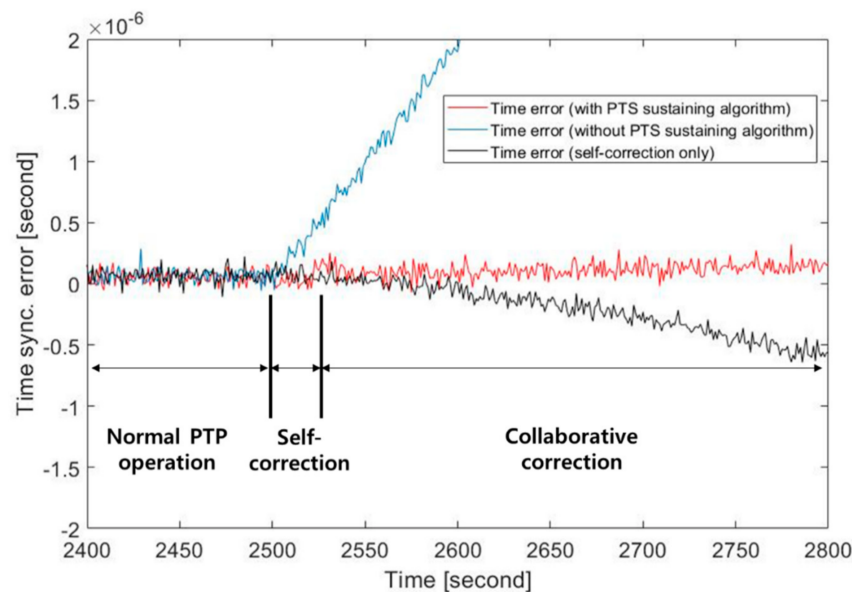


Figure 7. Comparison of time synchronization performance of precision time protocol (PTP) clocks.

Percent duration of PTS sustaining was observed for varying standard deviation of clock offset to evaluate PTS sustaining performance of the proposed algorithm. Target sustaining duration was set to 10 min. The simulation was performed 500 times and the percent duration was obtained by averaging the simulation results. Fifty clocks participated in estimating the collaboration-based reference clock. The results of the simulation are summarized in Table 1. It is confirmed that the performance of the PTP clock with the proposed algorithm surpasses the performance of the normal PTP clock.

Table 1. Comparison of percent duration of PTS sustaining of PTP clocks.

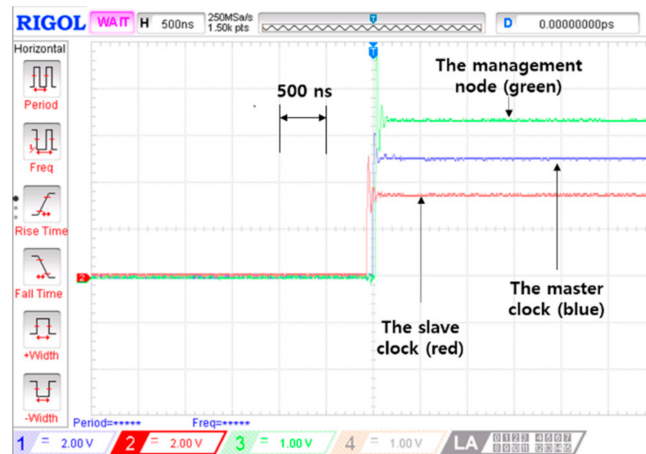
Standard Deviation of Clock Offset [kHz]	Percent Duration of PTS Sustaining [%]	
	Without PTS Sustaining Algorithm	With PTS Sustaining Algorithm
1	62.05	100.00
5	20.81	77.82
7	14.82	69.10
10	10.32	57.19
15	6.17	41.63
20	5.53	28.76

3.2. Operation Test on Embedded DSP System

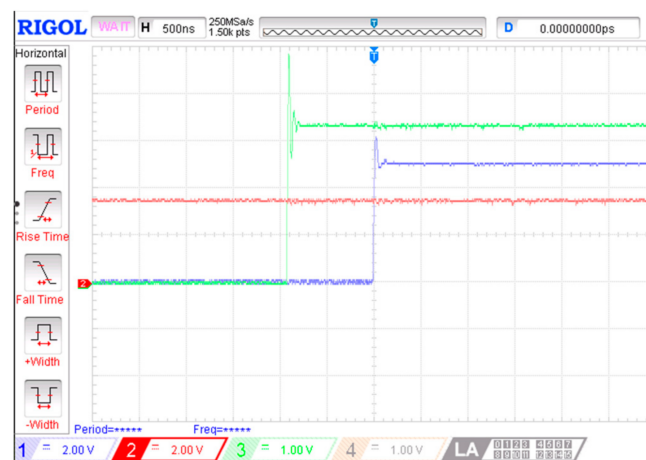
The proposed algorithm was implemented with TMD5DK572 multi-processor boards to illustrate the operation of the proposed method. The oscillator of the board is a normal crystal oscillator which has 30 ppm of clock accuracy at 20 MHz. The Linuxptp reference source code [21] was used to enable basic PTP functions.

Two EVM boards and PTP equipment were used for the self-correction operation test. The two EVM boards behaved as slave clocks during the normal PTP operation. One of the two EVM boards was preselected as a management node. A LANTIME M600 high end time server was used as a grandmaster clock and an RSG 2488 PTP supported network switch was used as a transparent clock. 1-PPS (pulse per second) signals of the EVM boards and the grandmaster clock were used to compare PTS sustaining performance (i.e., maintaining sub-microsecond precision). The master failure was simulated by pulling out the connected ethernet cable.

The observation results of 1-PPS signals during the master failure are shown in Figure 8. Three 1-PPS signals (i.e., the management node, the slave clock without the proposed algorithm and the grandmaster clock) were captured by an oscilloscope. The 1-PPS signal of the master clock was used as a reference signal in comparing PTS sustaining performances. When the master failure occurred, both the management node and the slave clock maintained sub-microsecond precision as shown in Figure 8a. The 1-PPS signal of the slave clock exceeds a 1 microsecond time difference after 20 s. On the other hand, the management node sustains a 1 microsecond time difference for 124 s which is very long time compared with the slave clock's performance, as shown in Figure 8b.



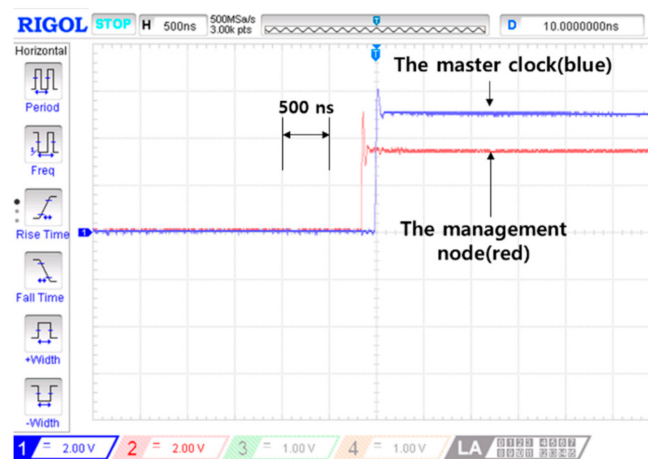
(a)



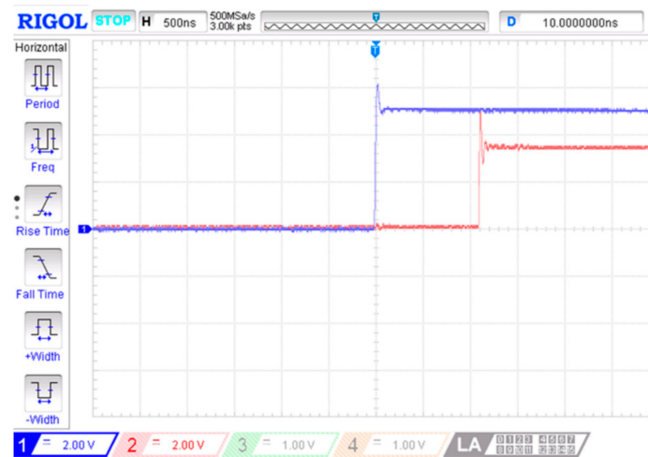
(b)

Figure 8. Operation test result of the proposed self-correction. Three 1 pulse per second(PPS) signal comparisons over time after the master failure. (a) 0 s (at the time of the master failure occurrence); and (b) 124 s.

The proposed collaborative PTS sustaining algorithm was implemented with three EVM boards and the PTP equipment to confirm the operation of the proposed algorithm including the collaborative correction phase. Because of difficulty accessing the commercially provided PTP equipment, propagation delay compensation is indirectly reflected by pre-calculated delay values among the EVM boards. The example of PTS sustaining obtained from the experimental test system is illustrated in Figure 9. It is observed that the collaborative correction sustains more than 1 min which is expected to prolong as the number of slave clocks increases. It is worth mentioning that major implementation elements of the proposed PTS sustaining algorithm (i.e., the timestamping and delay correction of the management packets) are readily realizable with the availability of directly modifiable PTP equipment.



(a)



(b)

Figure 9. Operation test result of the proposed collaborative PTS sustaining algorithm. Two 1-PPS signal comparisons over time after the master failure. (a) 0 s (at the time of the master failure occurrence); and (b) 62 s.

4. Conclusions

This paper proposes a PTS sustaining method which prolongs sub-microsecond time synchronization precision upon master failure. PTS sustaining is achieved by the successive application of a self-correction and a collaborative correction algorithm. The self-correction performs the prediction of clock drift characteristics and the collaborative correction gathers clock information of slave nodes and estimates a reference clock for the management node-based PTP synchronization. The proposed method tries to maximize PTS sustaining after master failure by using slave clocks having relatively low clock quality. The computer simulation results verified that PTS sustaining performance of the proposed algorithm applied to a PTP clock surpasses the performance of a normal slave clock. The operability of the proposed algorithm was validated from the embedded DSP hardware operation test. The operation test also showed that the proposed algorithm sustains sub-microsecond precision for 124 s after master failure. This is considered a very long time compared with the PTS sustaining performance of the normal slave clock, which is 20 s. The proposed algorithm is considered a software-based solution which can provide PTS sustaining means upon the total failure of a master clock and backup equipment. Therefore, the proposed PTS sustaining method is expected to provide a significant contribution for the reliable operation of industrial applications.

Proper clock grouping can provide enhanced accuracy in the estimation of a reference clock. The design of the clock grouping, which is left to future research, requires in-depth studies on various system-related factors including sustaining PTP specifications, network topology and traffic, system configuration and clock behavior modeling.

Author Contributions: Conceptualization, K.J.S.; writing-original draft, K.J.S.; software, K.J.S.; visualization, K.J.S.; methodology, T.G.C.; writing-review & editing, T.G.C.; supervision, T.G.C. Both authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Korea Electric Power Corporation, grant number R17XA05-2.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. IEEE. 1588™—IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems; IEEE: New York, NY, USA, 2008. [\[CrossRef\]](#)
2. Maftai, D.; Bartos, R.; Noseworthy, B.; Carlin, T. Implementing Proposed IEEE 1588 Integrated Security Mechanism. In Proceedings of the IEEE International Symposium on Precision Clock Synchronization for Measurement Control, and Communication, Geneva, Switzerland, 30 September–5 October 2018. [\[CrossRef\]](#)
3. Yin, H.; Fu, P.; Qiao, J.Q.; Li, Y.P. The implementation of IEEE 1588 clock synchronization protocol based on FPGA. In Proceedings of the IEEE International Instrumentation and Measurement Technology Conference, Houston, TX, USA, 14–17 May 2018. [\[CrossRef\]](#)
4. Cui, S.J.; Zeng, P.; Wang, L.; Li, P.; Huang, X.T.; Xu, Q. Research on Time Synchronization Technology for Wide-Area Measurement in the Distribution Network. In Proceedings of the International Conference on Information Science and Control Engineering, Zhengzhou, China, 20–22 July 2018. [\[CrossRef\]](#)
5. Dong, M.; Qiu, Z.L.; Pan, W.T.; Chen, C.; Zhang, J.X.; Zhang, D. The Design and Implementation of IEEE 1588v2 Clock Synchronization System by Generating Hardware Timestamps in MAC Layer. In Proceedings of the International Conference on Computer, Information and Telecommunication Systems, Colmar, France, 11–13 July 2018. [\[CrossRef\]](#)
6. IEEE. C37.238-2011—IEEE Standard Profile for Use of IEEE 1588 Precision Time Protocol in Power System Applications IEEE; IEEE SA: New York, NY, USA, 2011. [\[CrossRef\]](#)
7. IEEE. IEEE C37.238-2017—Standard Profile for Use of IEEE 1588 Precision Time Protocol in Power System Applications; IEEE SA: New York, NY, USA, 2017. [\[CrossRef\]](#)
8. IEEE. 61850-9-3—IEC/IEEE Precision Time Protocol Profile for Power Utility Automation; IEEE SA: Geneva, Switzerland, 2016. [\[CrossRef\]](#)
9. Son, K.J.; Chang, T.G. The Effect of Time Synchronization Error in LAN-based Digital Substation. *Sensors* **2019**, *19*, 2044. [\[CrossRef\]](#) [\[PubMed\]](#)
10. Bi, T.; Guo, J.; Xu, K.; Zhang, L.; Yang, Q. The impact of time synchronization deviation on the performance of synchrophasor measurements and wide area damping control. *IEEE Trans. Smart Grid* **2017**, *8*, 1545–1552. [\[CrossRef\]](#)
11. Zhang, Z.; Gong, S.; Dimitrovski, A.D.; Li, H. Time synchronization attack in smart grid: Impact and analysis. *IEEE Trans. Smart Grid* **2013**, *4*, 87–98. [\[CrossRef\]](#)
12. Moussa, B.; Debbabi, M.; Assi, C. A Detection and Mitigation Model for PTP Delay Attack in an IEC 61850 Substation. *IEEE Trans. Smart Grid* **2018**, *9*, 3954–3965. [\[CrossRef\]](#)
13. Yang, Q.; An, D.; Yu, W. On time desynchronization attack against IEEE 1588 protocol in power grid systems. In Proceedings of the IEEE Energytech, Cleveland, OH, USA, 21–23 May 2013; pp. 1–5. [\[CrossRef\]](#)
14. Seo, Y.D.; Son, K.J.; An, G.S.; Nam, K.D.; Chang, T.G.; Kang, S.H. Improved Time-Synchronization Algorithm based on Direct Compensation of Disturbance Effects. *Sensors* **2019**, *19*, 3499. [\[CrossRef\]](#) [\[PubMed\]](#)
15. Giorgi, G.; Narduzzi, C. Performance Analysis of Kalman-Filter-Based Clock Synchronization in IEEE 1588 Networks. *IEEE Trans. Instrum. Meas.* **2011**, *60*, 2902–2909. [\[CrossRef\]](#)
16. Lv, S.; Lu, Y.M.; Ji, Y.F. An Enhanced IEEE 1588 Time Synchronization for Asymmetric Communication Link in Packet Transport Network. *IEEE Commun. Lett.* **2010**, *14*, 764–766. [\[CrossRef\]](#)
17. Lee, S.; Lee, S.; Hong, C. An Accuracy Enhanced IEEE 1588 Synchronization Protocol for Dynamically Changing and Asymmetric Wireless Links. *IEEE Commun. Lett.* **2012**, *16*, 190–192. [\[CrossRef\]](#)

18. Gaderer, G.; Loschmidt, P.; Sauter, T. Improving fault tolerance in high-precision clock synchronization. *IEEE Trans. Ind. Inform.* **2010**, *6*, 206–215. [[CrossRef](#)]
19. Kozakai, Y.; Kanda, M. Keeping clock accuracy on a master clock failure in substation network. In Proceedings of the IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication, Portsmouth, NH, USA, 27 September–1 October 2010; pp. 25–29. [[CrossRef](#)]
20. IEC. IEC-62439-3—*Industrial Communication Networks—High Availability Automation Network—Part 3: Parallel Redundancy Protocol and High-Availability Seamless Redundancy (HSR)*; IEC: Geneva, Switzerland, 2012. [[CrossRef](#)]
21. The Linux PTP Project. Available online: <http://linuxptp.sourceforge.net/> (accessed on 7 August 2020).



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).