A Novel Method for Providing Precise Time Synchronization in a Distributed Control System Using Boundary Clock

Jae Hun Cho, Hoon Kim, Sihai Wang, Jaehoon Lee, Hanlim Lee, Seongtaek Hwang, Sungdae Cho, Yunje Oh, and Tae-Jin Lee, Member, IEEE

Abstract—We propose and demonstrate a method to provide precise time synchronization in distributed control systems using a boundary clock scheme. The major drawback of the boundary clock scheme is the exponential accumulation of time synchronization error as the number of hops increases. To make the error accumulation linearly increase with the number of hops, we first separate the frequency compensation interval (FCI) and the offset and frequency compensation interval (OFCI) and then separately optimize each interval. To demonstrate the performance of this method, we implemented test benches using Ethernet-linked distributed control systems. We measured the peak-to-peak jitter performance along with the maximum time interval error to assess the short- and long-term stability after several hops in distributed control systems. Our method enables the peak-to-peak jitter to be maintained under 107 ns after seven hops. The experimental results show that the performance of time synchronization is dominated by fast jitter rather than frequency error and wander, and the proposed scheme can be used to improve the time synchronization performance in IEEE 1588-compliant control systems using boundary clock.

Index Terms—Boundary clock, IEEE 1588, network synchronization, transparent clock, unshielded twisted pair (UTP).

I. INTRODUCTION

In Ddistributed control systems, where multiple control nodes are connected with communication links, there is a growing interest in providing time synchronization to nodes with time-sensitive control applications. To meet the interest, IEEE 1588 standards define time and message formats for the time synchronization of distributed control systems [1], [2]. In the standards, the time synchronization information carried over the boundary clock and/or the transparent clock periodically propagates from a grand master node to slave nodes.

The boundary clock is a clock that has multiple precision time protocol (PTP) ports in a domain. It may serve as the source of time (i.e., a master clock) and synchronize another clock, which becomes a slave clock. In the boundary clock scheme, therefore, the grand master first synchronizes the adjacent slave nodes. The slave nodes that are now synchronized with the grand master node then act as a master node to other adjacent slave nodes and synchronize them. IEEE 1588 standard version 1 exploits this scheme for time synchronization [1].

The transparent clock can be classified into two types: end-to-end transparent clock and peer-to-peer transparent clock. They are both specified in IEEE 1588 standard version 2 draft [2]. The end-to-end transparent clock supports the use of the end-to-end delay measurement mechanism between the slave and master clocks. The slave nodes located between the grand master node and a slave node, which is to be synchronized, pass through the time synchronization information between the two nodes, which makes it transparent to the information flow. Thus, every slave node is directly synchronized by the grand master node in this scheme. Due to the logical point-to-point connection between the grand master and slave nodes, however, this scheme might give rise to network congestion, and, thus, it might not be a good scheme for bandwidth-hungry applications. The other transparent clock scheme, i.e., peer-to-peer transparent clock, provides the PTP event transit time information as well as the correction information on the link propagation delay to the port receiving the PTP event message.

One of the merits of this scheme is that, unlike the end-to-end counterpart, it does not induce network congestion around the grand master node. However, the peer-to-peer transparent clock is inexistent in IEEE 1588 standard version 1, wherein only the boundary clock scheme is defined.

Recently, some researchers have proposed and demonstrated IEEE 1588-compliant precise time synchronization schemes [3]–[7]. However, their demonstrations are limited to single-hop experiments and might not be used for multihop distributed control systems, which require high accuracy in time synchronization. The major drawback of the boundary clock scheme is...
the accumulation of time synchronization error as the number of hops increases. According to the simulation study in [8], for example, the peak-to-peak error of clock synchronization is measured to be 0.8, 8, 40, and 300 μs for one, three, five, and ten hops, respectively. Thus, this exponential behavior of time synchronization error could be a stumbling block to the scalability of distributed control systems using the boundary clock scheme. In this paper, we propose and demonstrate a method to provide precise time synchronization in multihop distributed control systems using the boundary clock scheme. We also show that even if the frequency compensation algorithm per node does not guarantee the linear decrease in the accumulation of synchronization error, the frequency compensation algorithm together with offset compensation can be used for high-precision distributed control systems. To make the accumulation of error to linearly increase with the number of hops, we first separate the frequency compensation interval (FCI) and the offset and frequency compensation interval (OFCI), and separately optimize the intervals. Our experimental demonstration shows that the time synchronization error indeed linearly increases with the slope of 10.3 ns/hop, thus achieving less than 107 ns of peak-to-peak time synchronization error after seven hops. This result exhibits significant improvement over the recent demonstration of cascaded IEEE 1588-enabled switches, which presents the peak-to-peak jitter performance of 1000 and 3000 ns for four- and nine-hop links, respectively [9]–[12].

II. PROPOSED TIME SYNCHRONIZATION SCHEME

The IEEE 1588 standard defines the time and message protocol to deliver the time information of the master node to slave nodes but it does not define what to do with the information. In particular, the standard leaves the implementation of algorithms to achieve time synchronization open.

Recently, Balasubramanian et al. have proposed and demonstrated a frequency compensation algorithm for precise time synchronization using the IEEE 1588 protocol [13]. Based on the time information obtained from the IEEE 1588 synchronization process, they first calculate the frequency compensation value using their algorithm and reduce the frequency offset between the master and slave nodes by adjusting a counter in the slave node. In a single-hop experiment, they were able to achieve a worst-case time deviation of +/−100 ns on a switched 100-Mb/s Ethernet link. However, the frequency compensation algorithm per node does not guarantee the linear growth of synchronization error with the number of hops. According to our experimental result, the frequency compensation algorithm proposed in [13] exhibits an exponential accumulation of synchronization error as the number of hops increases (see Fig. 6 for details).

The accumulation of synchronization error can substantially be reduced by using offset compensation together with frequency compensation. The synchronization error linearly increases with time when there is frequency error between the master and the slave. Therefore, frequency compensation is very important to achieve accurate synchronization. However, frequency compensation per se is vulnerable to unexpected offset errors, which can possibly arise from the circuit noise and time-varying transmission delay. These can be corrected by using periodic offset compensation. The periods for frequency compensation and offset compensation should separately be optimized to minimize the synchronization error. Since the optimum periods are dependent upon the implementation methods and devices, we optimize the compensation interval through experiments.

Fig. 1 shows the ladder diagram of the proposed time synchronization algorithm. The master and slave nodes carry out the time synchronization process in a periodic manner with $t_{of}$ of OFCI and $t_f$ of FCI. The values of $t_{of}$ are set to an integer multiple of $t_f$ to make the FCI periodic.

For each synchronization process, we follow the IEEE 1588 standard: the master node first sends the Sync message to a slave node. An optional Follow_Up message can immediately be followed to the slave node to notice the sending time ($T1$) of the Sync message. The slave node then sends the Delay_Request message to the master node. Upon receiving the message, the master node records the arriving time ($T4$) of the message and generates the Delay_Resp message containing $T4$ to send it to the slave node. Since the slave node knows the sending time of the Delay_Req message (i.e., $T3$), the propagation time between the master and slave nodes can be easily calculated based on $T3$ and $T4$. Thus, the slave device can calculate $T1$, $T2$, $T3$, and $T4$ through this procedure, and the time offset value relative to the master device $T_{offset}$ can be expressed as

$$T_{offset} = \left( \frac{[T2 - T1] - (T4 - T3)}{2} \right).$$

(1)

The offset and frequency compensation values obtained from the proposed algorithm illustrated in Fig. 1 are used to generate an offset- and frequency-compensated clock. A local oscillator in the slave node increases the clock counter at every cycle of the oscillator. However, since the master clock runs at a different pace with the slave’s local oscillator, the output of the clock counter in the slave node should be compensated based on the time synchronization process.

The details of the offset and frequency compensation algorithm are as follows. As illustrated in Fig. 1, the master transmits a Sync message to the slave at $MasterSyncTime_n$, where $n$ is the count number of the Sync message. The Sync (Follow_Up) message contains the time value $MasterSyncTime_n$ obtained by time stamping. The slave receives the message at its own local clock time of $SlaveClockTime_n$. The slave then computes the master clock time $MasterClockTime_n$, which can be expressed as [13]

$$MasterClockTime_n = MasterSyncTime_n + MasterToSlaveDelay$$

(2)

where $MasterToSlaveDelay$ corresponds to a transmission delay, which should be updated when there is a change in the network configuration. The master clock count $MasterClockCount_n$ of the current synchronization cycle can be expressed as

$$MasterClockCount_n = MasterClockTime_n - MasterClockTime_{n-1}.$$  

(3)
Likewise, the slave clock count $\text{SlaveClockCount}_n$ for the current synchronization cycle can be written as

$$\text{SlaveClockCount}_n = \text{SlaveClockTime}_n - \text{SlaveClockTime}_{n-1}. \quad (4)$$

The difference of clock count between the master and the slave for the current synchronization cycle $\text{ClockDiffCount}_n$ can then be found as

$$\text{ClockDiffCount}_n = \text{MasterClockTime}_n - \text{SlaveClockTime}_n. \quad (5)$$

The compensation process is performed by determining whether to increase/decrease or hold the counter value, which depends upon the offset and frequency compensation values. The offset and frequency compensation value obtained from the $n$th time synchronization process $\text{CompValue}_n$ can be defined as

$$\text{CompValue}_n = \text{ScaleFactor}_n \times \text{CompValue}_{n-1}. \quad (6)$$

In our proposed scheme, $\text{ScaleFactor}_n$ can be expressed as either frequency scale factor $\text{FreqScaleFactor}$ or offset and frequency scale factor $\text{OffsetFreqScaleFactor}$, depending upon which process it belongs to.

For the FCI process, $\text{FreqScaleFactor}$ at the $n$th time synchronization process can be expressed as

$$\text{FreqScaleFactor}_n = \frac{\text{MasterClockCount}_n}{\text{SlaveClockCount}_n} \quad (7)$$

where $\text{MasterClockCount}_n$ is the time difference between two successive time synchronization processes measured in the master node, and $\text{SlaveClockCount}_n$ is the time difference between two successive time synchronization processes measured in the slave node.

For the OFCI process, $\text{OffsetFreqScaleFactor}_n$ at the $n$th time synchronization process can be written as

$$\text{OffsetFreqScaleFactor}_n = \frac{\sum_{i=n-\gamma+1}^{n} \text{MasterClockCount}_i}{\sum_{i=n-\gamma+1}^{n} \text{SlaveClockCount}_i} + \frac{\text{ClockDiffCount}_n}{\text{SlaveClockCount}_n} \quad (8)$$

where

$$\gamma = \frac{t_{of}}{t_{df}}. \quad (9)$$

It should be noted that although the FCI is fixed in our proposed scheme, it does not imply that the Delay_Req time inside each sync process is also fixed. IEEE 1588 version 1 specifies a mandatory randomization of Delay_Req messages to spread the master’s load [1]. In our scheme, the Delay_Req time (i.e., $T_3$) can be randomized within a sync process since it does not affect the accuracy of time synchronization.

III. EXPERIMENTAL DEMONSTRATION AND RESULTS

To demonstrate the performance of the proposed scheme, we implemented four sets of test benches using a 10/100/1000-Mb/s Ethernet PHY (Marvell, 88E1141), a field-programmable gate array (FPGA) with dual PPC405 processors (Xilinx, SC2VP40), a synchronous dynamic random access...
memory (SDRAM), and Flash memories. A photograph of the implemented test bench is shown in Fig. 2. Each test bench runs on a 125-MHz clock signal, which is generated by doubling the output of a 62.5-MHz crystal oscillator with 50-ppm frequency accuracy. For the fair comparative study between our scheme and Balasubramanian’s scheme, we utilized the same physical hardware. The only difference between the two schemes is the algorithm implemented with hardware description language (HDL) and C language, which is ported to the FPGA and processor from the computer.

Since each set is capable of supporting two separate master/slave nodes, we configured a daisy-chain link with eight nodes indexed 0–7, as illustrated in Fig. 3. We set Node 0 to be the grand master. Since we employ a boundary clock scheme, Node \( i \) becomes a master node to slave Node \((i + 1)\).

Fig. 4(a) and (b) shows the captured oscilloscope images after one and seven hops, respectively. The oscilloscope is triggered with a rising edge of 8-kHz pulse signal that is generated at the grand master. We measured the rising edge of the 8-kHz pulse signal at each slave node. The peak-to-peak jitter of each slave node is measured by tracing the rising edge of the signal at a digital storage oscilloscope with the bandwidth of 6 GHz. Since the peak-to-peak performance can depend on the measurement time, we set the measurement interval to be 1 h.

Before we measure the performance of the implemented test bench, we first determine the best value of \( \gamma \), as in (9), which is the ratio of OFCI to FCI by experiments. Fig. 5 shows the measured peak-to-peak jitter at slave nodes as a function of the number of hops. To change \( \gamma \), we set OFCI to be 1 s and changed FCI. This is because a long OFCI is beneficial to bandwidth utilization, and we found that an OFCI of 1 s does not sacrifice the jitter performance. The measured results in Fig. 5 show that the optimum \( \gamma \) is 8, which corresponds to an FCI of 125 ms.

Fig. 6 shows the measured peak-to-peak jitter as a function of the number of hops when \( \gamma = 8 \). In Balasubramanian’s scheme, the frequency compensation algorithm is set to be 125 ms for the sake of fair comparison with the proposed scheme. As shown in the figure, the peak-to-peak jitter increases in an exponential manner as the number of hops increases. For example, the peak-to-peak jitter after one hop is measured to be 95 ns. However, after seven hops, it is increased to 40 \( \mu s \). In addition, depicted in Fig. 6 is the jitter performance of the proposed scheme. The proposed scheme exhibits a linear hop-number-versus-jitter performance. The peak-to-peak jitters are measured to be 44.55 and 106.36 ns for one and seven hops, respectively. It should be noted that, compared to Balasubramanian’s scheme, the proposed scheme reduces the peak-to-peak jitter by half even at the single-hop experiment.

To assess the long-term stability of the proposed scheme, we measured the maximum time interval error (MTIE) of the pulse signal. The results are depicted in Fig. 7. In this figure, the MTIE values monotonically increase with the observation...
window. The starting value (i.e., MTIE value at the smallest observation window) implies fast jitter characteristics. The long-term frequency stability can be observed with the slope of the MTIE curves. The measured curves in Fig. 7 show a shallow slope of the MTIE regardless of the number of hops, which implies that the long-term frequency error is very small, and that the peak-to-peak jitter performance in Fig. 6 is mainly determined by fast jitter rather than slow wander.

Fig. 8 shows the peak-to-peak jitter performance when 1-m unshielded twisted pair (UTP) cables are replaced with 25-m UTP cables. This is to show the robustness of the proposed scheme when the propagation delay is long (> 900 ns). The total length of the UTP cables spanning seven hops is 175 m. For comparison purposes, the result in Fig. 6 is also reproduced in this figure. The figure shows the high robustness of the proposed scheme against various cable lengths. No performance degradation is observed even with 25-m UTP cables. Although the UTP cable length of 25 m/hop might not be long enough to represent real-world systems, we believe that the experimental results at least exhibit the robustness of our scheme against propagation delay.

IV. CONCLUSION

We have proposed and demonstrated a method to provide precise time synchronization in distributed control systems using a boundary clock. Basically, the proposed scheme complies with the IEEE 1588 standard. However, to improve the jitter performance in a large-scale distributed control system, we utilize fast frequency compensation together with slow offset compensation. The proposed scheme is demonstrated through an experiment by using Ethernet-linked distributed control systems. The system consists of Ethernet links, an FPGA with processors, an SDRAM, and Flash memories. We measured the peak-to-peak jitter performance along with the MTIE to assess the short- and long-term stability. The peak-to-peak jitter was measured to be < 107 ns even after seven hops. The experimental results also show that the performance of time synchronization is dominated by fast jitter rather than frequency error and wander. The experiment performed with different lengths of UTP cables confirms the robustness of the scheme against propagation delay. Therefore, we believe that the proposed scheme can be used to improve the time synchronization performance in IEEE 1588-compliant control systems using boundary clock.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their constructive comments and suggestions.

REFERENCES

Hoon Kim, photograph and biography not available at the time of publication.

Sihae Wang, photograph and biography not available at the time of publication.

Jae Hun Cho received the B.S. and M.S. degrees in electronics engineering from Kyunghee University, Suwon, Korea, in 2000 and 2002, respectively. Since 2002, he has been with Samsung Electronics Company, Suwon, Korea, where he is currently a Senior Engineer with the Telecommunication R&D Center, and worked on the development of precise time synchronization in distributed control systems and radio over fiber (RoF) technology for beyond third generation (B3G) mobile communication systems. He is also with the School of Information and Communication Engineering, Sungkyunkwan University, Suwon. His current research interests include multimedia signal processing for the mobile terminal.

Hanhim Lee received the B.S., M.S., and Ph.D. degrees in radio engineering from Korea University, Seoul, Korea, in 1997 and 2003, respectively, and the M.S. degree in electrical engineering from the Korea Advanced Institute of Science and Technology, Taejon, Korea, in 1999. From 2003 to 2008, he was with the Telecommunication R&D Center, Samsung Electronics, Suwon, Korea. Since 2008, he has been an Assistant Professor with the Department of Computer and Communication Engineering, Korea University. His current research interests include the modeling and simulation of optical and microwave components and high-speed transmission systems.

Seongtaek Hwang received the B.S., M.S., and Ph.D. degrees in physics from Sogang University, Seoul, Korea, in 1988, 1990, and 2001, respectively, for his work on optical communication. Since 1993, he has been with Samsung Electronics, Suwon, Korea, where he has been working on the development of optical-fiber amplifier, subsystem for optical transmission system application, fiber-to-the-home (FTTH) system, and radio over fiber (RoF) technology for beyond third-generation (B3G) mobile communication systems. His current research interests include optical character recognition (OCR) for the mobile terminal and motion and gesture recognition technology for mobile terminal applications.

Cho Sungdae received the B.S. degree in computer science from Soongsil University, Seoul, Korea, in 1996 and the M.S. and Ph.D. degrees from Rensselaer Polytechnic Institute (RPI), Troy, NY, in 2000 and 2002, respectively.

After two years of postdoctoral research work with the Center for Image Processing Lab, RPI, in 2004, he joined Samsung Electronics Company, Suwon, Korea, where he is currently a Senior Engineer with the Telecommunication R&D Center. His research interests include multimedia signal processing, color processing, and graphics applications on mobile systems.

Yunje Oh received the B.S. and M.S. degrees in physics from Yonsei University, Seoul, Korea, in 1985 and 1987, respectively, and the Ph.D. degree in physics from Rensselaer Polytechnic Institute, Troy, NY, in 1995, for his work on nonlinear optics and optical solitons.

Since 1996, he has been with Samsung Electronics, Suwon, Korea, where he has been working on the development of optical-fiber communication systems, optical switch systems, and optical-access systems. His current research interests include residential Ethernet, radio over fiber (RoF), and multimedia and mobile imaging.

Tae-Jin Lee (M’00) received the B.S. and M.S. degrees in electronics engineering from Yonsei University, Seoul, Korea, in 1989 and 1991, respectively, the M.S.E. degree in electrical engineering and computer science from the University of Michigan, Ann Arbor, in 1995, and the Ph.D. degree in electrical and computer engineering from the University of Texas, Austin, in May 1999.

In 1999, he was a Senior Engineer with the Corporate R&D Center, Samsung Electronics, Seoul, Korea. Since 2001, he has been an Associate Professor with the Pennsylvania State University, University Park. His research interests include performance evaluation, resource allocation, medium access control (MAC), and design of communication networks and systems, wireless LAN/PAN/MAN, home/ad-hoc/sensor/RFID networks, next-generation wire-less communication systems, and optical networks.

Dr. Lee has been a voting member of the IEEE 802.11 WLAN Working Group since 2004.