

A Practical Implementation of IEEE 1588-2008 Transparent Clock for Distributed Measurement and Control Systems

Jiho Han, *Student Member, IEEE*, and Deog-Kyoon Jeong, *Senior Member, IEEE*

Abstract—This paper addresses issues with time synchronization using the IEEE 1588-2008 for distributed measurement and control systems. A practical implementation of the transparent clock is presented with the overall system architecture and detailed operation of each building block. To verify the submicrosecond accuracy using the implemented devices, an experimental setup that was analogous to a practical distributed system has been built. Measured results from the experiment show that the time error is limited below 30 ns for nodes that were connected by three switches. It is remarkable that the results are observed in spite of large packet queuing delays that were introduced by a traffic generator. The discussion on sources of time error that was outlined here provides technical considerations to designing IEEE 1588 systems.

Index Terms—Distributed systems, Ethernet, IEEE 1588, networked measurement, synchronization, transparent clock (TC).

I. INTRODUCTION

SYNCHRONIZATION has been an important issue in various applications. In particular, in distributed measurement and control systems, the overall performance greatly depends on the synchronization accuracy between each node over the network. As a means of providing robust cost-effective time synchronization for the distributed systems, the IEEE 1588 [1] specifies the precision-time protocol (PTP). Recent research effort has been focused on applying the IEEE 1588 to practical measurement and control systems [2]–[4]. The fundamental idea of the IEEE 1588 is very close to that of the network-time protocol (NTP) [5] based on packet exchanges, but the target accuracy is much higher (on the order of submicroseconds) [6]. By taking timestamps on the boundary of the physical layer for each network interface, the time error can be reduced below 100 ns for a very simple experimental setup.

For a larger network of practical distributed systems, however, the target accuracy cannot be achieved without considering the packet queuing delays. Data traffic may cause long delays (on the order of milliseconds), because the same network resources are shared with IEEE 1588 messages. Although

filtering techniques may be helpful in reducing the effect of the queuing delays, they require extremely stable clock sources that will be equipped in every slave for submicrosecond accuracy. As a solution to this problem, the latest version of the IEEE 1588 defines a transparent clock (TC). A network element that operates as a TC measures the residence time that the IEEE 1588 messages experience inside the network element itself. Each IEEE 1588 slave then eliminates the error that results from queuing delay by using the residence-time information on the messages, which is updated by the network elements.

Related works have been published as the TC captures growing attention in many applications. Mohl *et al.* [7] were the first researchers to publish a paper on TC design issues. They implemented and characterized TCs and measured the performance over seven cascaded TCs. Innicki *et al.* [8] describe the performance of a TC prototype that works over existing routers and switches. Köhler [9] presents a practical implementation of an IEEE 1588 that supports Ethernet switch. Ferrari *et al.* [10] deal with synchronization of the probes of a distributed instrument system. Their probes are realized using a field-programmable gate array (FPGA) to run IEEE 1588 PTP and the Precision TC Protocol (the proprietary protocol). Garner *et al.* [11] present simulation results that show that an ideal network that supports the IEEE 802.1AS (a Layer-2 profile of the IEEE 1588-2008) meets many synchronization requirements. Early implementation of IEEE 802.1AS bridges and the performance test are described by Teener *et al.* [12]. Eidson *et al.* [13] discuss a device, i.e., the spider TC, which allows a few ports of an ordinary bridge or router to act as a TC. Meier *et al.* [14] describe an IEEE 1588 clock that completely realizes synchronization and synchronization functions in the hardware. Their implementation combines a three-port bridge with peer-to-peer TC functionality and an ordinary clock (OC).

Each paper explains the basic operation and design issues of the TC. The performance measurement that was shown in most papers, however, is conducted in simple networks with only two or three devices. Moreover, IEEE 1588 messages do not experience large queuing delays at the network switches, because no data traffic or only the traffic with short frames is applied in the synchronization network. To verify the essential function of a TC, it has to be shown that submicrosecond accuracy can be achieved under the condition that the message exchanges are disturbed by large queuing delays. It is also important to build a network that is large enough to model the realistic distributed measurement and control systems.

Manuscript received November 12, 2008; revised March 17, 2009. The Associate Editor coordinating the review process for this paper was Dr. Jesús Ureña.

The authors are with the School of Electrical Engineering and Computer Science, Seoul National University, Seoul 151-742, Korea, and also with the Interuniversity Semiconductor Research Center, Seoul National University, Seoul 151-742, Korea (e-mail: jhhan@mystic.snu.ac.kr; dkjeong@snu.ac.kr).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIM.2009.2024371

This paper addresses the implementation and performance issues of the IEEE 1588-2008 TC. The major functions that were proposed in the new version are implemented in a Gigabit Ethernet switch. One prototype of the proposed switch architecture has been fabricated into a 21-mm² silicon chip. To verify the accuracy improvement by the TC, an experimental setup that was analogous to the practical measurement and control system is designed. According to the result of performance evaluation, the time error is not more than 30 ns between the IEEE 1588 master and any slave over the test setup with 50% traffic load. It is a remarkable improvement, considering that the time error (on the order of tens of microseconds) is observed without TCs.

The rest of this paper is organized as follows. Section II presents a brief overview on the major features of the IEEE 1588-2008. In the next two sections, detailed description on the proposed architecture, design, and implementation is followed by the experimental setup and measured data using the implemented TCs. Section V discusses the sources of time error to provide technical guidelines for higher accuracy. This paper concludes with the summary and contribution.

II. IEEE 1588-2008

The IEEE 1588 is a standard for a precision clock synchronization protocol for networked measurement and control systems. The standard specifies a protocol for time synchronization and supporting communication properties. Although the basic idea is similar to that of NTP based on packet exchanges with timestamps, the IEEE 1588 mandates dedicated hardware support for much higher accuracy. Assisted by the synchronization circuit, every network interface takes very accurate timestamps and manages precisely synchronized time. According to the standard, the IEEE 1588 has the following four distinguished features:

- 1) spatially localized;
- 2) microsecond to submicrosecond accurate;
- 3) administration free;
- 4) accessible for both high-end and low-cost low-end devices.

The IEEE 1588-2008 has recently been developed to meet synchronization needs in new applications. After the first IEEE 1588 was standardized in 2002, the more refined applications in industrial automation, test and measurement, and military suggested the following new requirements:

- 1) higher accuracy;
- 2) varied update rates;
- 3) linear topology (in addition to hierarchical topology);
- 4) rapid reconfiguration after network changes;
- 5) fault tolerance.

Furthermore, new applications in telecommunications, the IEEE 802.1AS (audio/video bridging network), and the power industry are an additional driving force of the discussion on the new version.

Several new features are proposed in the IEEE 1588-2008 to meet the aforementioned objectives. For higher accuracy, a correction field of 64-bit integer is introduced, which consists of a 48-bit nanosecond part and a 16-bit subnanosecond part.

The correction field eliminates a large portion of calculation error, because the least significant bit (LSB) represents 2^{-16} ns. For varied update rates, a SYNC message interval that is as short as 4 ms is allowed, whereas the original IEEE 1588 specifies the minimum interval of SYNC messages to 1 s. In addition, the TC is defined to address the issue of long linear topologies. IEEE 1588 devices that support the peer-delay mechanism measure a peer delay for every communication link to rapidly reconfigure synchronization hierarchy. Moreover, the new version defines ANNOUNCE messages for the selection of the best master clock. By splitting the information for master selection and timing, very short SYNC messages can deliver timing information with minimized consumption of network resources. According to the proposed short-message format, the length of the SYNC message payload is 46 B, which is only a quarter of 165 B in the original standard. The listed features improve the synchronization performance and flexibility of the IEEE 1588 for application in several areas.

III. IMPLEMENTATION ISSUES

The proposed synchronization hardware for supporting the IEEE 1588-2008 has been integrated inside a prototype of the Gigabit Ethernet switch. After several tests using an FPGA device, the switch architecture has been fabricated into a 21-mm² silicon chip using the 0.18- μ m complementary metal oxide semiconductor (CMOS) technology. The chip is an experimental application-specific integrated circuit, which includes an IEEE 1588 circuit, medium access control (MAC), and switch fabric, as shown in Fig. 1(a). Commercial 1000BASE-T PHYs of different vendors (e.g., National Semiconductor and Marvell) are attached outside the implemented CMOS device. They show different latencies, which are the important sources of time error as will be explained. Such vendor-specific errors are calibrated to remove the static time error. The IEEE 1588 circuit delivers services of both the TC and the OC. If the switch works as a pure TC, it is sufficient to measure the residence time and peer delay without time synchronization. The OC function is implemented for the application where the switch itself needs to maintain the synchronized time. As a peer-to-peer TC, the implemented prototype switch forwards the received SYNC messages after updating the correction field by measuring peer delay and residence time.

Fig. 1(a) describes the structure and operations of building blocks. The IEEE 1588 circuit is implemented right over the MAC layer of each port interface. It consists of a TX unit, an RX unit, and a CLOCK unit, which operate as follows. First, the TX unit generates SYNC, FOLLOW_UP, PDELAY_REQ, PDELAY_RESP, and PDELAY_RESP_FOLLOW_UP messages as defined in the IEEE 1588-2008. Each message contains the specified information, e.g., the timestamps that were provided by the CLOCK unit or the calculated values. Second, the RX unit deals with IEEE 1588 messages to extract the necessary information. First, it determines if the received Ethernet frame contains an IEEE 1588 message and what kind of message it is. The RX unit then parses each field of the message and calculates the period and offset data using the extracted data. The `offset_gm` and `period_gm` are the time and frequency errors

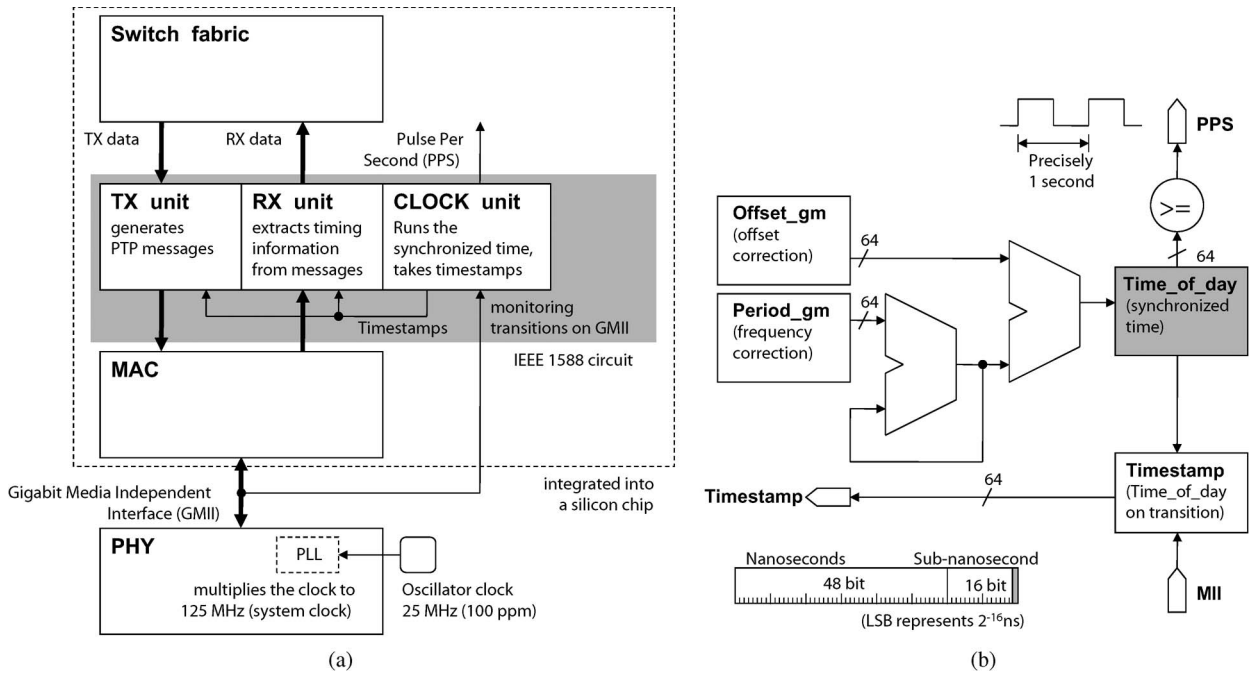


Fig. 1. (a) IEEE-1588-compliant switch. (b) Operation of the CLOCK unit.

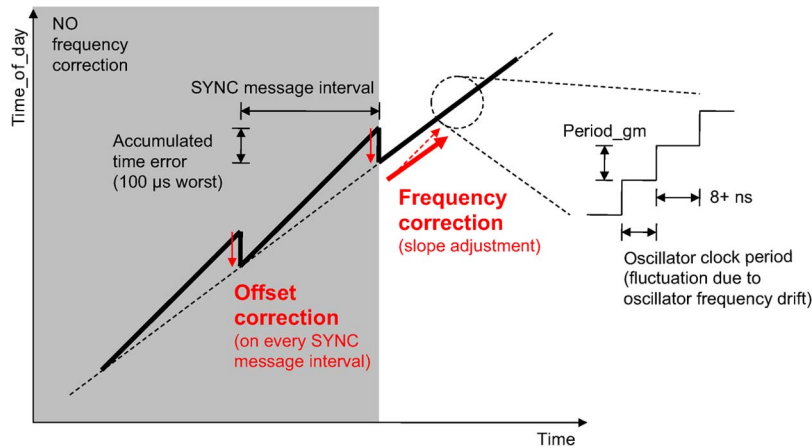


Fig. 2. Relation between time_of_day and the physical time.

of the slave from the IEEE 1588 master, respectively. Third, the CLOCK unit takes timestamps and runs the synchronized time. The detailed operations of the CLOCK unit are shown in Fig. 1(b). As time advances, the period_gm that was passed from the RX unit is accumulated every system clock cycle. Summation of the accumulated value and the offset_gm gives the time_of_day, which is the synchronized time itself. Taking timestamps at the right moment is another important function of the CLOCK unit. It is recommended to take timestamps when the end of the start-of-frame delimiter is detected on the gigabit media-independent interface (GMII). By saving the time_of_day during detection, the CLOCK unit generates 64-bit values. For an output interface to other timing equipment, pulse per second (PPS) is generated by asserting a pulse whenever the time_of_day is equal to every second.

For submicrosecond accuracy, frequency correction is one of the most essential features in the CLOCK unit. It is assumed that the typical frequency stability of the local oscillators in

Ethernet devices is ± 100 ppm. Although most Ethernet devices use clocks with a tolerance of 100 ppm as defined in the IEEE 802.3, the performance improvement is expected with the use of better oscillators (30 ppm) that have recently been available due to improved wireless transceivers (e.g., IEEE802.15.4 and WirelessHART [15]). Each local oscillator has its own frequency near the nominal value. Although the difference seems to be negligible, it may grow up to hundreds of microseconds after accumulation for a few seconds. Fig. 2 displays this accumulation by plotting the time_of_day versus the physical time. At first, the time_of_day of the IEEE 1588 master increases along the broken line, whose slope is 1 under the assumption that the ideal master keeps the exact physical time. Consider the case when the local oscillator of a slave runs faster than the master by 100 ppm. If both the master and the slave simply accumulate 8 every clock cycle, which is 8 ns, the slave will have about 1 000 100 000 after 1 s. The correct time, however, is 1 000 000 000 as indicated in the master. The slave obtains

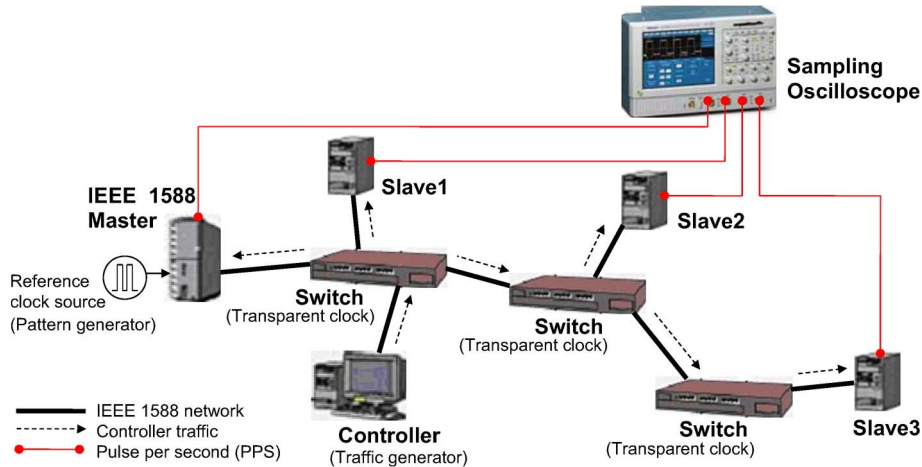


Fig. 3. Experimental setup that models a practical measurement and control system.

timing information from the master on every SYNC message interval; thus, the larger the time deviation is, the longer the message interval becomes. After the slave receives a SYNC message from the IEEE 1588 master, the `time_of_day` of the slave is relocated on the broken line. As described in Fig. 2, the `time_of_day` of the slave increases along the thick solid line, whose slope is 1.0001. Before the slave receives a SYNC message, the worst time error of $100 \mu\text{s}$ occurs. To eliminate the large error, the slope of the thick line must be correct to 1 by accumulating 7.9992 instead of 8. The CLOCK unit calculates the exact frequency from the ratio between time differences in the master and the slave for the same SYNC messages.

In Fig. 2 (right), the frequency error is shown as corrected; thus, the `time_of_day` of the slave increases along the broken line. After much of the frequency error is corrected, a small error still exists because of the frequency drift of the unstable oscillator. Zooming in the circle, the `time_of_day` of the slave time advances with discrete jumps along the step function rather than smoothly increasing along a straight line. Although the height of a single step is fixed, the time interval varies with time, because oscillator characteristics are affected by thermal noise and supply voltage fluctuation. The amount of such error depends on the short-term stability or the period jitter of local oscillators. According to the electrical specification and measured results of the oscillator that we used, the root-mean square period jitter is much less than 100 ps.

IV. EXPERIMENTAL RESULTS

An experimental setup in Fig. 3 has been built to verify the performance improvement by using TCs. The setup is designed to model a practical distributed measurement and control system with a master node, three slave nodes, three switches, and a controller. Three IEEE-1588-2008-compliant Gigabit Ethernet switches are used to build a simple local area network that connects the five devices. The nodes work as OCs that were implemented in FPGA-based prototype systems, which deliver the very OC functions in Section III and use exactly the same IEEE 1588 circuits as those in the switches. The master node has accessibility to a very accurate clock source, which is a pattern generator in this experiment. On the other hand, slaves use

crystal oscillators and synchronize to the master by exchanging messages through the network. To measure the synchronization accuracy, PPS signals of the four nodes are connected to a high-speed sampling oscilloscope, as shown in Fig. 3. The controller generates several types of dummy traffic to introduce random queuing delays in the switches. Although the communication in practical systems is much more complicated, simple artificial traffic is used in the experiments to see the adverse effect of queuing delays. The controller broadcasts Ethernet frames with a fixed packet length of 64, 256, and 1518 B to all slaves with a traffic load of 50%. There is no packet loss in the test setup in Fig. 3. In the case where packet loss occurs in a more practical situation, some SYNC messages are dropped, and the slaves then less frequently receive the timing information. The performance degradation is expected to be equivalent to the case where the SYNC interval is increased, which would not cause any drastic effect. The real network for distributed measurement and control, however, is generally managed to keep the traffic load under 50% so that packet loss seldom occurs.

The SYNC message interval is 1 s, and the interval of peer delay measurements is 1 s. Although a PI control loop in each slave is effective in reducing the time error when TCs are not used, it is not as effective when using TCs. The reason is that it requires an extremely stable local oscillator to filter out the remaining error components after the time error due to the queuing delay being removed by TCs. In our experiments, filtering techniques are not used in calculating the synchronized time. Although the circuit can operate in either on-the-fly (SYNC without FOLLOW_UP) or two-step mode (SYNC followed by FOLLOW_UP), the two-step mode is used in the measurement, which is more generally used.

Our experimental results suggest that the implemented TCs play an important role in achieving submicrosecond accuracy. An oscilloscope screenshot in Fig. 4 shows the distribution of the time error. The four channels display PPS's of the IEEE 1588 master, slave1, slave2, and slave3. They are sampled for 10 min, with the master PPS being selected as a trigger source. As shown in Fig. 4, rising edges of the PPS of a slave wander within each interval around zero. Detailed statistical data of the time error for each slave are summarized in Table I. Note that

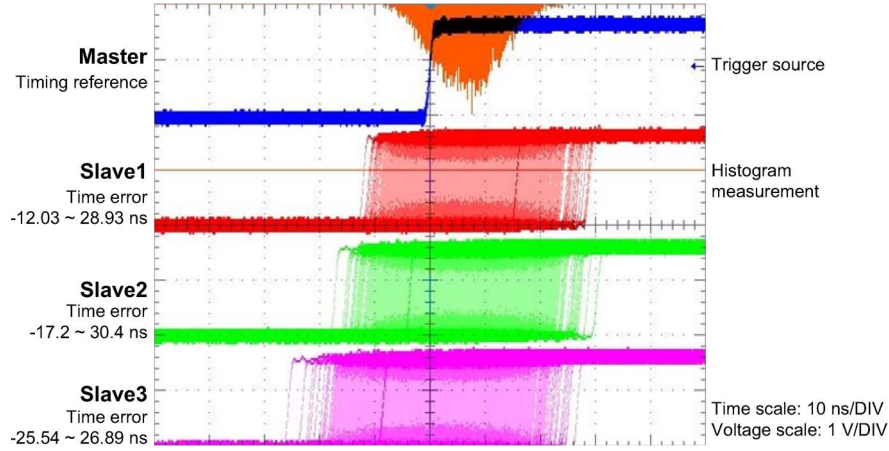


Fig. 4. Oscilloscope screenshot that measures time errors of the three slaves.

TABLE I
STATISTICAL DATA OF TIME ERROR USING TCs

	Mean	St. Dev.	Maximum
Slave1	6.7	5.9	29
Slave2	6.0	6.4	30
Slave3	1.5	6.8	27

All units in nanoseconds. Time errors are shown not to be affected by the traffic conditions since transparent clocks enable IEEE 1588 slaves to eliminate the time error caused by the queuing delay.

the maximum time error is not more than 30 ns for any slave. High accuracy (on the order of tens of nanoseconds) is shown to be feasible for a practical model of distributed measurement and control systems. For comparison with the methodology in the literature, the same configuration and topology is built. Most papers, including [8]–[10] and [13], measured the performance in the simple topology where two OCs are connected directly or through one TC. When background traffic is not applied, the referenced papers present measured errors of ± 80 ns, ± 90 ns, a maximum of 244 ns, and ± 80 ns, respectively. Mohl *et al.* [7] estimated a peak-to-peak error of 175 ns from a standard deviation of 25 ns. Under the same conditions, our system provides a time error between -12 and 29 ns in slave1, as shown in Fig. 4.

The performance improvement by using the implemented TCs is more obvious when compared with the results using ordinary switches. Similar experiments have been performed after the three TCs are replaced with commercial ordinary switches that have no support for the IEEE 1588. The switches that we selected for comparison are BCM5395 [16] and GS105 [17], which are five-port Gigabit Ethernet switches of Broadcom and Netgear, respectively. Table II summarizes the maximum time errors for several traffic conditions. When no traffic is applied, the nodes could share synchronized time with submicrosecond accuracy. When the controller generates dummy traffic, however, the time error becomes much larger. In particular, in the case of 1518-B packets, the error is longer than 30 μ s.

TABLE II
MAXIMUM TIME ERROR USING ORDINARY SWITCHES

	No traffic	64B traffic	256B traffic	1518B traffic
Slave1	248	512	2,379	12,320
Slave2	354	856	3,679	23,610
Slave3	504	1,516	5,861	36,200

All units in nanoseconds. Maximum time errors are measured for the traffic conditions that are generated by the controller. Traffic conditions tested are: no traffic, Ethernet frames with the fixed length of 64, 256, and 1518 bytes.

When a PTP message is ready to transmit in a switch, the other dummy packet of various lengths may be in the middle of being transmitted. In this case, the PTP message is queued until the ongoing transmission ends. Therefore, the maximum queuing delay is equal to the serialization delay for the dummy packet. Maximum time errors in the table look the same as the serialization delay, because the errors resulted from asymmetric delays. Considering that the most common lengths of Ethernet frames are 64 and 1518 B, severe performance degradation by long packets may quite frequently occur. The implemented TCs measure the residence time to enable slave nodes to correct the time error due to the queuing delay. For this reason, the time error is not affected by the traffic condition in the IEEE 1588 network using TCs.

V. SOURCES OF TIME ERROR

In addition to the implementation and performance evaluation, sources of time errors are examined for further improvement. Fig. 5 shows a simple network where an IEEE 1588 master and a slave exchange messages through a switch. Note that the network shows a general example, regardless of whether a TC is used. First, asymmetry of the propagation delays explains a large portion of the time error. An IEEE 1588 slave estimates the propagation delay of SYNC messages by assuming that the delay is symmetric for both directions. This assumption, however, is not generally true over the network.

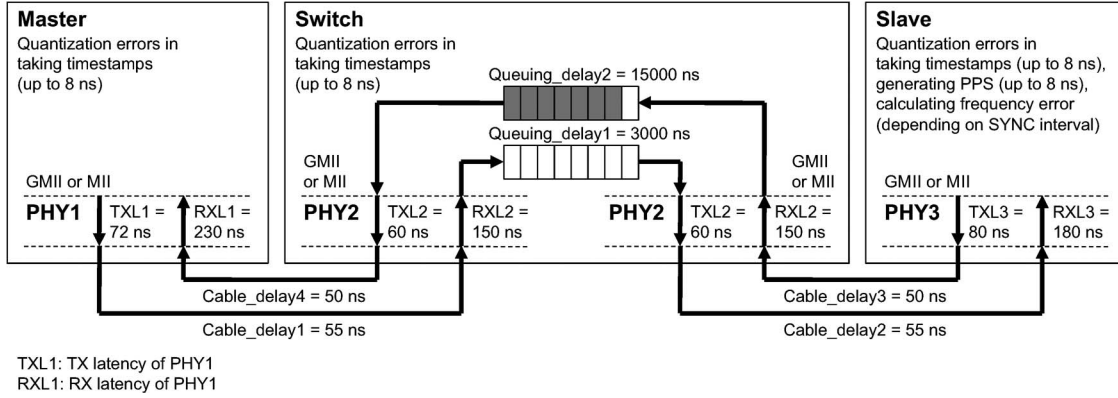


Fig. 5. Example routing path of synchronization packets between a master and a slave connected by a switch. The total amount of asymmetry can reach 11 948 ns along the thick lines, excluding quantization errors.

For each SYNC message, one IEEE 1588 circuit takes timestamp when the message is transmitted from MAC to PHY, and the other IEEE 1588 circuit in the opposite side of the Ethernet link takes a timestamp when the message is received from PHY to MAC. In measuring the time interval between the two moments of taking timestamps, errors occur at the PHY level due to the asymmetry of the propagation delay. The propagation delay consists of PHY transmit latency, cable delay, and PHY receive latency, as shown in Fig. 5. In Fast Ethernet (100BASE-TX) connection using unshielded twisted pair (UTP) cables, different pairs are used for each direction. Two different pairs are likely to have different wire lengths, and hence, the propagation delays are asymmetric. UTP cables can cause a time error (on the order of tens of nanoseconds) of more than 100 m. On the other hand, Gigabit Ethernet (1000BASE-T) uses all four pairs for both directions; thus, the asymmetry in cable delay is almost eliminated. However, the Gigabit Ethernet PHY uses extensive digital signal processing techniques that add significant latency and cause a few cycles of uncertainty in the sampling time. Furthermore, its latency significantly varies among PHY chip vendors due to different implementation styles. For the case shown in Fig. 5, PHY1 has a TX latency of 72 ns and an RX latency of 230 ns. PHY2 and PHY3 have their own latencies, as shown in Fig. 5. The asymmetries that were caused in the PHY layer have been dealt with as demonstrated in previous works [18], [19]. Measurement and correction can eliminate the time errors that were caused by vendor-specific PHY latency and asymmetric cable delay, because they are fixed or changed within relatively small ranges. In our experiments, every IEEE 1588 slave includes such correction capability to reduce the time error. Although it is vendor specific, the average latency can easily be corrected, and its variation of as much as a few tens of nanoseconds is more difficult to cancel out.

Queuing delay in the switch occupies the most significant part of asymmetry issues, because it may cause large variation in the propagation delay (on the order of milliseconds). When a message is forwarded from the master to the slave, as shown in Fig. 5, it takes 3000 ns in the switch. On the other hand, as an example of extreme asymmetry, a message from the slave to the master experiences a very long delay of 15 000 ns. To sum up the elements described so

far, the propagation delay in each direction is calculated as follows:

Delay_from_master_to_slave

$$\begin{aligned}
 &= \text{TXL1 (TX_latency_of_PHY1)} \\
 &\quad + \text{Cable_delay1} + \text{RXL2 (RX_latency_of_PHY2)} \\
 &\quad + \text{Queuing_delay1} + \text{TXL2 (TX_latency_of_PHY2)} \\
 &\quad + \text{Cable_delay2} + \text{RXL3 (RX_latency_of_PHY3)} \\
 &= 72 + 55 + 150 + 3000 + 60 + 55 + 180 \\
 &= 3572 \text{ ns}
 \end{aligned}$$

Delay_from_slave_to_master

$$\begin{aligned}
 &= \text{TXL3 (TX_latency_of_PHY3)} + \text{Cable_delay3} \\
 &\quad + \text{RXL2 (RX_latency_of_PHY2)} + \text{Queuing_delay2} \\
 &\quad + \text{TXL2 (TX_latency_of_PHY2)} + \text{Cable_delay4} \\
 &\quad + \text{RXL1 (RX_latency_of_PHY1)} \\
 &= 80 + 50 + 150 + 15\,000 + 60 + 50 + 230 \\
 &= 15\,620 \text{ ns.}
 \end{aligned}$$

The overall asymmetry in Fig. 5 is 12 048 ns, which is the time error of the experiment using ordinary switches. When using TCs, however, the queuing delays have no influence, because TCs measure and compensate for the exact residence time. If the switch has TC capability, the asymmetric queuing delays are corrected; hence, the time error is reduced to 48 ns. As aforementioned, by measurement and correction of PHY latencies and cable delays, the remaining asymmetry issues can almost be eliminated.

Quantization error means the difference between the actual value and the digitized value caused by finite operation frequency or finite word length. The CLOCK unit detects signal transitions on the GMII based on its own system clock to take timestamps. Whenever the IEEE 1588 message crosses a clock boundary where the signaling clock and the sampling clock are different, a nondeterministic error occurs. The maximum error is equal to the period of system clock for data sampling. Another quantization error rises when the IEEE 1588 device generates the PPS signal. The rising edge of the actual PPS signal can be 8 ns late from the exact moment to rise, because the CLOCK unit compares the time_of_day with target values

every 8 ns. Thus, the time difference between the master's PPS edge and the slave's PPS edge on the oscilloscope screen includes a random error of 8 ns. It would be technically precise to regard the PPS error as a measurement error rather than a synchronization error. This error can be eliminated if we use a phase-locked loop (PLL) or delay-locked loop that generates clock phases with a time resolution of less than 8 ns. Aside from the time-domain factors, a finite number of digits of digital computation also cause a quantization error. Although every slave corrects the frequency error by computing the ratio of the master's frequency to its own frequency, the quotient of division still has a quantization error within its LSB. Accumulation of this negligible error until the next SYNC message arrives brings a time error of a few nanoseconds.

In addition to asymmetry issues and quantization errors, oscillator stability and PLL internal jitter may cause a small amount of additional time error. As aforementioned, inexpensive crystal oscillators have low-frequency stability and random jitter affected by the supply voltage fluctuation and the thermal noise. The system clock that was provided by the PHY chip may generate an additional error, because the clock and data recovery circuit in each PHY has jitter (on the order of nanoseconds).

VI. CONCLUSION

This paper has presented the implementation and performance issues of the IEEE 1588-2008 TC. To verify the accuracy improvement by the TC, the experimental setup analogous to the practical measurement and control system is designed. Based on the experimental results, the maximum time error is less than 30 ns for the node that was connected by three TCs. It is remarkable that the synchronization performance is maintained in spite of large queuing delays. In addition to the performance evaluation, this paper has provided an investigation of error sources for further improvement.

ACKNOWLEDGMENT

The authors would like to thank H. Chi for his contribution, Dr. G. Garner, Dr. C. Lee, Dr. E. Ryu, Dr. B. Kim, Dr. P. Jeong, and Dr. F. Feng of Samsung Electronics for their support and discussion, the IC Design Education Center (IDEC), Korean Advanced Institute of Science and Technology, Daejeon, Korea, for chip fabrication, and the anonymous reviewers for their valuable advice.

REFERENCES

- [1] *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*, IEEE Std. 1588-2008, 2008.
- [2] P. Ferrari, A. Flammini, D. Marioli, and A. Taroni, "IEEE-1588-based synchronization system for a displacement sensor network," *IEEE Trans. Instrum. Meas.*, vol. 57, no. 2, pp. 254–260, Feb. 2008.
- [3] L. Barford and J. Burch, "Fourier analysis from networked measurements using time synchronization," *IEEE Trans. Instrum. Meas.*, vol. 56, no. 5, pp. 1601–1604, Oct. 2007.
- [4] T. Cooklev, J. C. Eidson, and A. Pakdaman, "An implementation of IEEE 1588 over IEEE 802.11b for synchronization of wireless local area network nodes," *IEEE Trans. Instrum. Meas.*, vol. 56, no. 5, pp. 1632–1639, Oct. 2007.

- [5] D. L. Mills, "Internet time synchronization: The network time protocol," *IEEE Trans. Commun.*, vol. 39, no. 10, pp. 1482–1493, Oct. 1991.
- [6] L. De Vito, S. Rapuano, and L. Tomaciello, "One-way delay measurement: State of the art," *IEEE Trans. Instrum. Meas.*, vol. 57, no. 12, pp. 2742–2750, Dec. 2008.
- [7] D. Mohl and H. Weibel, "Prestandard prototype implementation of an end-to-end transparent clock," in *Proc. Conf. IEEE 1588*, Oct. 2006. [Online]. Available: <http://ieee1588.nist.gov/>
- [8] S. Ilnicki, T. Hidai, and J. Burch, "Performance of transparent clock over existing network elements using modified pluggable gigabit interface," in *Proc. Conf. IEEE 1588*, Oct. 2006.
- [9] D. Köhler, "A practical implementation of an IEEE 1588 supporting Ethernet switch," in *Proc. IEEE Int. Symp. Precision Clock Synchronization Meas., Control Commun.*, Oct. 2007, pp. 134–137.
- [10] P. Ferrari, A. Flammini, D. Marioli, S. Rinaldi, and A. Taroni, "Synchronization of the probes of a distributed instrument for real-time Ethernet networks," in *Proc. IEEE Int. Symp. Precision Clock Synchronization Meas., Control Commun.*, Oct. 2007, pp. 33–40.
- [11] G. M. Garner, F. Feng, K. Hollander, H. Jeong, B. Kim, B. Lee, T. Jung, and J. Jeong, "IEEE 802.1 AVB and its application in carrier-grade Ethernet," *IEEE Commun. Mag.*, vol. 45, no. 12, pp. 126–134, Dec. 2007.
- [12] M. D. Johas Teener and G. M. Garner, "Overview and timing performance of IEEE 802.1AS," in *Proc. IEEE Int. Symp. Precision Clock Synchronization Meas., Control Commun.*, Sep. 2008, pp. 49–53.
- [13] J. C. Eidson, A. Fernandez, B. Hamilton, J. Naous, and D. Vook, "Spider transparent clock," in *Proc. IEEE Int. Symp. Precision Clock Synchronization Meas., Control Commun.*, Sep. 2008, pp. 7–11.
- [14] S. Meier, H. Weibel, and K. Weber, "IEEE 1588 syntonization and synchronization functions completely realized in hardware," in *Proc. IEEE Int. Symp. Precision Clock Synchronization Meas., Control Commun.*, Sep. 2008, pp. 1–4.
- [15] *Wireless HART Data Sheet*, HCF, Austin, TX, May 15, 2007.
- [16] *5-Port GbE Switch With Five Integrated PHYs and Packet Buffer Product Brief*, Broadcom, Irvine, CA, May 2, 2007.
- [17] *ProSafe 5-Port and 8-Port Gigabit Ethernet Desktop Switches 10/100/1000 Mbps Data Sheet*, Netgear, San Jose, CA, Jun. 27, 2008.
- [18] C. Thurnheer, J. Blattner, H. Weibel, and M. Rupf, "Determination of the IEEE 1588 relevant timing behavior of 100 base-TX PHYs," in *Proc. Conf. IEEE 1588*, Oct. 2005.
- [19] T. Müller, A. Ockert, and H. Weibel, "PHYs and symmetrical propagation delay," in *Proc. Conf. IEEE 1588*, Sep. 2004, pp. 78–84.



Jiho Han (S'08) was born in Seoul, Korea, in 1979. He received the B.S. and M.S. degrees in electrical engineering in 2002 and 2004, respectively, from Seoul National University, Seoul, where he is currently working toward the Ph.D. degree in electrical engineering and computer science with the School of Electrical Engineering and Computer Science and the Interuniversity Semiconductor Research Center.

His research interests include time synchronization using the IEEE 1588, carrier-class Ethernet, and design of integrated systems.



Deog-Kyoon Jeong (S'87–M'89–SM'09) received the B.S. and M.S. degrees in electronics engineering from Seoul National University, Seoul, Korea, in 1981 and 1984, respectively, and the Ph.D. degree in electrical engineering and computer sciences from the University of California, Berkeley, in 1989.

From 1989 to 1991, he was with Texas Instruments, Inc., Dallas, as a Member of Technical Staff, working on the modeling and design of BiCMOS gates and the single-chip implementation of the Scalable Processor Architecture. He joined the faculty of

the Department of Electronics Engineering and the Interuniversity Semiconductor Research Center, Seoul National University. From 2006 to 2008, he was the Director of ISRC. He is one of the Cofounders of Silicon Image, which specializes in digital interface circuits for video displays such as digital visual interface and high-definition multimedia interface. He has published more than 60 technical papers and is the holder of 52 U.S. patents. His research interests include the design of high-speed I/O circuits, phase-locked loops, and network switch architectures.