Proceeding of the 2015 IEEE
International Conference on Information and Automation
Lijiang, China, August 2015

Time Synchronization Method of IEEE 802.1AS through Automatic Optimal Sync Message Period Adjustment for In-Car Network

Yong Ju Kim, Bo Mu Cheon, Jin Ho Kim, Jae Wook Jean
Department of Electrical and Computer Engineering
Sungkyunkwan University
Suwon, Republic of Korea
dragonju@skku.edu, cheonbomu@naver.com, besteng83@gamil.com, jwjeon@yurim.skku.ac.kr

***Abstract* - We investigated the performance of the IEEE 802.1AS time synchronization protocol and its usage in in-vehicle networks by assessing the precise time synchronization mechanisms, such as offset correction, rate correction and the automatic optimal sync message period. In particular, this paper presents a method for adjusting the optimal synchronization message period to achieve high-precision time synchronization for the in-car network. Since the in-vehicle network is a static network, and performs synchronization when the ignition is turned on, the synchronization algorithm is not required for a certain period of time. Performing the synchronization algorithm too frequently can lead to vehicle malfunction and an increase in the network traffic load. Thus, according to current in-vehicle network circumstances, the best synchronization performance of the in-vehicle network is proposed by determining an optimum synchronization message period.**

Index Terms* - *Ethernet AVB, IEEE 802.1AS, Time synchronization, Optimal sync message period, in-vehicle network.

I. INTRODUCTION

The use of communications networks to connect in-vehicle electronic control units (ECU) dates back to the beginning of the 1990s [1]. Modern vehicles can contain up to 70 electronic control units (ECUs) of different functionalities, and these are interconnected through specific automotive protocols, such as LIN, CAN, FlexRay, MOST, and TTCAN [2], [3]. While CAN and FlexRay were designed to service real-time applications with small bandwidth demands, MOST was designed to provide a higher bandwidth that still fulfills real-time requirements, and it is used to provide synchronous transport of audio and video streams in automotive multimedia and infotainment systems [4]. However, the automotive industry is currently developing other communications solutions to reduce cable and connector costs by allowing the use of more flexible architectures, such as Ethernet [5].

As a result, Ethernet is expected to be the dominant next generation in-vehicle network protocol [6], and an Ethernet audio/video bridge (AVB) will then be used for infotainment terminals and multimedia applications. The use of ECUs and infotainment functions in automotive systems has been increasing, and in particular, Ethernet has been actively researched for use in in-vehicle networks. For example, the IEEE 802.1 AVB standard can guarantee data quality with real-time functionality [7]. The Ethernet AVB standard includes the IEEE 802.1AS time synchronization protocol, which provides a high-precision clock for synchronous streaming services. For these reasons, optimal performance of the time synchronization subsystem is essential in order to accurately synchronize each audio and video stream within the in-car network [8]. Since time synchronization is an important mechanism, many in-vehicle networks provide such functionality, including TTCAN [9], FTT-CAN [10], and FlexRay [11], [12] to support a combination of both time-triggered and event-triggered transmissions.

For various applications, packet transmission can be considered to be a representative example of time synchronization performed on an asynchronous network based on the IEEE 1588 PTP and IEEE 802.1AS standards. The IEEE 1588 standard is generally to synchronize time, and many studies have been conducted on this standard. However, research on IEEE 802.1AS for use in in-vehicle systems has been relatively limited. Although Lim et al. [4] discussed the synchronization performance of IEEE 802.1AS, the performance evaluation experiments consisted of simulations only. The time synchronization method in a common IEEE 1588 PTP network consists of determination of the reference time by executing a specific algorithm and synchronization of time by calculating the delay between the master and the slave. Common time synchronization techniques do not reflect the traffic conditions or the state of the slave nodes of the network. Therefore, when a large amount of traffic is generated in a network, time synchronization is adversely impacted. Systems that require strict determinism, such as Factory automation equipment, industrial networks and ADAS (Advanced Driver Assistance System) which belong to in-vehicle networks, have a number of slave nodes with a short correction cycle. Therefore, the time load for synchronization operation cannot be ignored. The amount of synchronization messages sent from the master to the slave, that is, the sync message transmission intervals, are initially set to a constant value. The time synchronization system, where the interval for transmitting sync messages is fixed, cannot reflect changes in the surrounding environment in real time, and it is possible for synchronization performance to deteriorate greatly. Time synchronization operations can generate the additional load. Synchronization requires not only accuracy in timekeeping, but also measures taken to reduce the load caused by the time synchronization operation. A general method to correct the timer involves calculating the offset in the time between the exchange of synchronization messages between the master and slave nodes. By using the value, it is possible to perform inter-node offset correction. Offset correction and rate correction mentioned above are methods known to be relevant to time

978-1-4673-9104-7/15/$31.00 ©2015 IEEE

synchronization. An in-car network is a network in which all ECUs are statically configured and do not change over their life-time. So, the node with the best clock inside of the in-car network (e.g., GPS-equipped devices) is set manually. The synchronization message period influences the timing error of the time-aware systems. A lower synchronization message period leads to better synchronization accuracy, but also to a higher network load. The synchronization interval has a significant influence on synchronization error and on accuracy [4]. This paper presents a method for adjusting the optimal synchronization message period for high-precision time synchronization in an in-car network. This method is proposed as the following: depending on the state of traffic state or the state of the slave nodes in the in-vehicle network, the number of sync messages to be transmitted from the master are automatically adjusted to minimize the traffic of the entire network in order to optimally maintain the synchronization performance of a node.

This paper is structured as follows. Section II provides essential, related work. Section III introduces the background of an AVB system. Section IV describes time synchronization for IEEE 802.1AS. Section V describes the proposed time synchronization method for an in-car network. Section VI provides the configuration of the experiment along with the experimental results of IEEE 802.1AS time synchronization. Section VII concludes our work.

II. Related Work

Garner et al. [13] discussed how to achieve accuracy for a generalized Precision Time Protocol (gPTP) in a Gigabit network with a maximum of 7 hops is below ±500ns. Their work addressed the accuracy of clock-synchronization over Audio/Video Bridging (AVB)-enabled bridges.

Lim et al. [4] implemented an AVB in an Ethernet-based in-car network to provide driver assistance and rear-seat entertainment features. They introduced an AVB model for in-car networks and used a network simulation to show that accuracy remained below 1μs, as required by IEEE 802.1AS.

Emanuel et al. [14] addressed the applicability of using AVB in a fully-switched Ethernet network that covers safety-related functions for aeronautics. The result of this work is the performance of audio transmission approaches in aeronautics where we address AVB without synchronizing AVB with IEEE 802.1AS.Time-sensitive applications are described in Garner et al. [15], [16] and the jitter, wander, and synchronization requirements are described in Garner et al. [17]. Kern et al. [18] discussed clock synchronization over two hops with a special focus on AVB synchronization in a varying temperature range. The results of this work were promising in terms of their accuracy, which ranged in the low double-digits (ns). Imtiaz et al. [19] discussed the use of AVB in industrial real-time communication and provided a comprehensive comparison between AVB and standard Ethernet.

III. AVB Background

TABLE I
Ethernet AVB Standards

Stack	Explanation
IEEE 802.1AS	Timing and synchronization protocol for time-sensitive applications [21]
IEEE 1722	Layer2/Layer3 transport protocol for time-sensitive applications (AVBTP) [22]
IEEE 802.1Qat	Stream reservation protocol [23]
IEEE 802.1Qav	Forwarding and queuing rules for time-sensitive applications [24]
IEEE 802.1BA	Profiles for Audio Video Bridging systems [25]

Audio/Video Bridging (AVB) is a technology standard that conforms to IEEE 802.1. It was designed to provide a smooth transmission to ensure the quality of multimedia stream data, such as audio and video data. AVB technology is a technique that can be used to synchronize clocks around an Ethernet Bridge to enable streaming services that can minimize transmission delays. It is required by Layer 2 in the OSI model, and it is necessary to build an AVB-based switch for network infrastructure. The media network corresponds to the existing CobraNet and EtherSound, and it has a latency of 2ms [7]. An Ethernet audio/video bridge (Ethernet AVB) is composed of a comprehensive set of specifications that provides high-precision clock streaming applications over 802.3 (Ethernet), including audio and video data [19]. The following table shows an Ethernet AVB standard stack [20].

IEEE 802.1AS [26] is a standard being developed in the 802.1 Working Group to provide high-precise timing and synchronization for bridged networks. In order to provide highly-precise time synchronization, IEEE 802.1AS uses a gPTP based on the IEEE 1588-2008 standard [27]. It is one of the 802.1 AVB standards, and the major use of it will be to make sure that jitter, wander, and time synchronization requirements for time-sensitive applications in in-vehicle networks are met [13].

IV. Time Synchronization Based on IEEE 802.1AS

A single node in an AVB network is defined as the grandmaster node that uses the best master clock algorithm (BMCA) to select the best clock within the gPTP domain. It exchanges *announce* messages with the nodes and identifies time-aware systems that support the IEEE 802.1AS protocol. Once it has been selected by using the BMCA, the

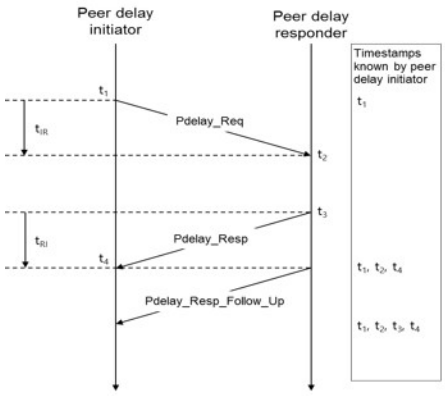

Fig 1. Propagation delay using a peer-delay mechanism

grandmaster node starts the synchronization process. The timing information that is sent by the grandmaster is received by the connected nodes, and they exchange synchronization information by adding the residence time and the propagation delay. This delay is measured by a peer-to-peer delay mechanism that is based on a peer delay request-response approach. Specifically, the peer-to-peer delay mechanism is distributed to the gPTP network by using three successive messages: *Pdelay_Req*, *Pdelay_Resp* and *Pdelay_Resp_Follow_Up* [28]. For the first step, the peer delay initiator initiates measurement by sending a *Pdelay_Req* message, while the responder responds with a *Pdelay_Resp* message that contains the t2 timestamp. In addition, a *Pdelay_Resp_Follow_Up* is sent from the responder to the initiator in order to transmit the t3 timestamp. For the final step, the peer delay initiator knows all timestamps and calculates the mean propagation delay using Equation (1), where r represents the clock drift of the responder.

Logical synchronization between the nodes in a network requires synchronization information with the residence time and propagation delay to be corrected. Since each of the nodes has different values for each crystal, the tick value for each node always has a time difference. The local clocks have a frequency variation, defined as a clock drift, that results in different time values between the master node and the slave nodes. Each node is a time-aware system that can be used to determine the clock drift of all nearby systems, which is defined as a *neighbor rate ratio*.

$$\text{pdelay} = \frac{(t4 - t1) - r * (t3 - t2)}{2} \qquad (1)$$

The synchronization information is exchanged to the gPTP network by using two successive messages: *Sync* and *Follow_Up*. The *Sync* and *Follow_Up* messages are sent periodically between the master and the slave ports in order to exchange synchronization information. Figure 2 shows the time synchronization process between three time-aware systems, where time-aware system A and time-aware system C are end nodes, and time-aware system B indicates a bridge. After the *Sync* message is sent from time-aware system A to time-aware system B, a *Follow_Up* message is transmitted with three pieces of information [26]:

- Precise origin time-stamp from the grandmaster
- Correction information updated by each time-aware system, including the propagation delay and the residence time.
- Frequency rate ratio relative to the grandmaster clock

These pieces of information are used to correct the synchronization information for each node. Finally, all time-aware systems are synchronized according to the IEEE 802.1AS protocol, and all time-aware systems are able to know the measured grandmaster rate ratio, where the clock drift between the grandmaster and the slave clocks is given.

V. Time Synchronization Algorithms Through Automatic Optimal Sync Message Period Adjustment

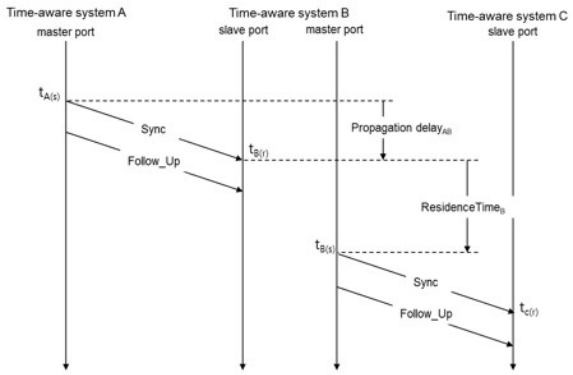

Fig 2. Synchronization process initiated by grandmaster

Generally, time synchronization involves using a technique that synchronizes the combined local time to the global time based on a timer provided by a remote device. Prior evaluation techniques for different network time synchronization protocols already exist, such as NTP (Network Time Protocol), SNTP (Simple Network Time Protocol), IEEE 1588 PTP, and IEEE 802.1AS. While NTP and SNTP do not meet the requirements for the high-precision accuracy of a clock's crystal oscillator and frequency [29], IEEE 1588 PTP and IEEE 802.1AS are able to perform more precisely.

An in-car network is a network in which all ECUs are statically configured and do not change over their life-times. So, the node with the best clock inside an in-car network is set manually. The synchronization message period influences the timing error of the time-aware systems, and a lower synchronization message period leads to better synchronization accuracy, but also to a higher network load. The synchronization interval has a significant influence on synchronization error and accuracy [4].

This paper presents a method for adjusting the optimal synchronization message period to provide high-precision time synchronization. With this method, depending on the state of the current network, the synchronization message period is automatically adjusted. Therefore, traffic on the network is minimized in order to maintain optimal synchronization performance. As a result of implementing the above method for in-car networks, a relatively static in-vehicle network can maintain the highest bandwidth possible. In an in-vehicle network, synchronization messages that are too frequently transmitted and received will result in a reduction in vehicle performance or another type of malfunction. The optimal synchronization message period is determined in order to minimize traffic so that the nodes of the in-vehicle network, including the ECUs, can provide the best possible synchronization performance.

A. *Offset correction*

Figure 3 shows the relationships between time and the time error between the master node and the slave node. The

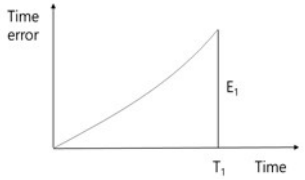

Fig 3. Time and time error relationship between nodes

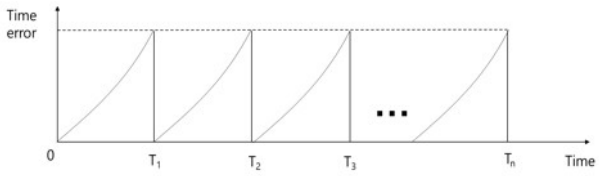

Fig 4. Time and time error relationship between the nodes after applying offset correction

description for Figure 3 is as follows. At time T_1, the time error between two nodes is caused by E_1. Over time, the time error value between the two nodes increases. Accordingly, attempts are made to improve the synchronization performance through a number of methods that reduce the time error.

Figure 4 shows the relationship between time and the time error when offset correction is applied. Although the time error value is 0 at first, it increases as times goes by. At time "T_1," after performing the synchronization algorithm to start sending the Sync message to the slave nodes from the master node, the time error approaches zero again. The time error value increases until the synchronization algorithm is performed once more. By repeating this process, a certain value for the time error between the master and slave nodes is not surpassed. It is then necessary to apply the rate correction function to further reduce the value of the time error between the master node and the salve node.

B. Rate correction

Figure 5 shows the relationship between time and time error when offset correction and rate correction are applied. The sync message period is the same as in Figure 4. Since offset correction and an additional rate correction are applied, the time error value decreases as the synchronization algorithm is carried out over time.

C. Optimal sync message period adjustment

Generally, the master node for the IEEE 1588 and IEEE 802.1as standards has a function to manually adjust the period of the sync messages that are sent to the slave node. Then, the value of the period of the given sync message is fixed until the synchronization algorithm is interrupted. Thus, the master node statically maintains the period of the synchronization message. The sync message period is 1 second long and can have a significant impact on the synchronization performance between the master node and the slave nodes. If the sync message period is short, the synchronization algorithm is executed frequently, and therefore, synchronization performance improves. However, when the synchronization message is transmitted too often, traffic increases on the network, and thus, synchronization performance decreases. On the other hand, if the synchronization message period is too long, synchronization performance can also decrease.

This paper therefore presents an algorithm that can adjust

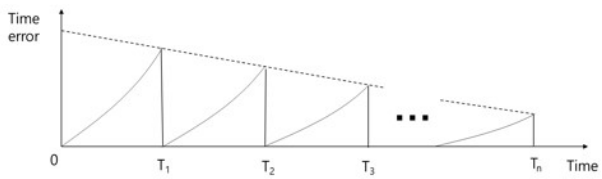

Fig 5. Time and time error relationship between the nodes after applying offset correction and rate correction

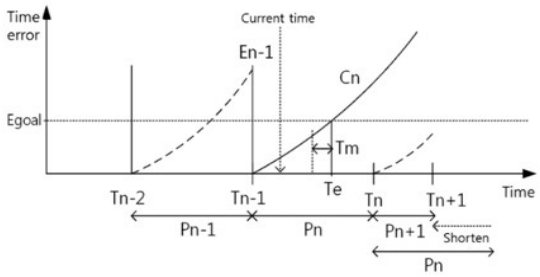

Fig 6. Example of the method used to determine the short synchronization message period

the period during which the synchronization message is sent from the master. Thus, the number of messages that are transmitted per unit of time is automatically controlled, and accordingly, can actively react to the environment to exert optimal synchronization performance. When the master begins to operate according to the reference period that was set during the initial synchronization, the message can be used to perform time synchronization with the slave. Depending on the number of synchronization messages that may be arbitrarily set and a variety of peripheral environmental conditions (such as the traffic state, the performance of the master and slave nodes, etc.), the most appropriate value must be established. For example, the calculation for the average number of proper synchronization messages is determined by the pre-treatment. Alternatively, it is possible to arrive at an intermediate value for the maximum number of messages that the master can support.

The algorithms presented in our paper are shown in Figures 6 and 7. When time synchronization occurs for a slave node in a row, the change in the error and the time offset are estimated according to the local change in time. The following graph shows an example of the method used to determine the synchronization message period. Figure 6 shows how the next time synchronization period (Pn+1) is determined during the current time synchronization period (Pn). The time error (En-1) is calculated using time synchronization operations performed during previous time synchronization periods (Pn-1) with an expiration date (Tn-1). The calculated time error (En-1) is used to estimate the time error curve (Cn), which in turn is used to determine the synchronization period for the next time (Pn+1) during the synchronization period for the current time (Pn). Since the present point in time is any point contained in the current time synchronization period (Pn), the expiration time (Tn) state has not yet reached. Therefore, the time synchronization error at the time of expiration for the

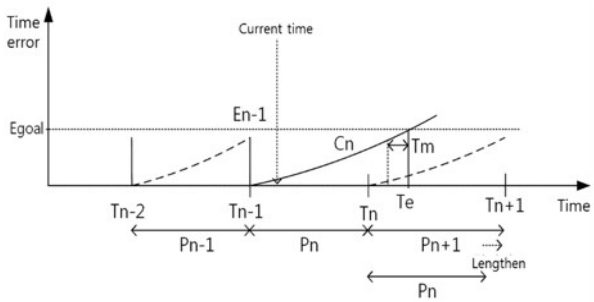

Fig 7. Example of the method used to determine the long synchronization message period

current time period (Pn) cannot be known in practice until the next synchronization operation is performed in accordance with the expiration time of the current time synchronization period (Pn). Synchronization period can be estimated from the current time. The calculated time error (En-1) obtained as a result of the previous time synchronization operations is used to estimate the time error curve (Cn). Accordingly, the time offset curve is estimated during the current time synchronization period (Pn), and a specified time error range is allowed (Egoal). Typically, the time is accurate to within 1 μs. The results of the comparison indicate that after the current time of the synchronization period (Pn) has expired, it is possible to determine the synchronization period (Pn+1) for the next time. The example in Figure 6 occurs when the allowable time difference range (Egoal) crosses the estimated time error curve (Cn). At the time on the estimated time error curve (Cn) that exceeds the range for the allowable time difference (Egoal) with excess time (Te), the excess time (Te) occurs before the current time synchronization period has expired. Assuming that the next time synchronization period has been determined (Pn+1) to be the same as the current time synchronization period (Pn) before the next time synchronization period expires (Pn+1), it is possible to estimate time error that exceeds the allowed time error range (Egoal). If the synchronization time period is not reduced, then the time error will consistently be larger than the allowed time error range (Egoal). Accordingly, the length of the next synchronization period (Pn+1) can be determined to be the same or less than the current time synchronization period (Pn) according to the excess time (Te). The length of the next time synchronization period (Pn+1) can be determined using four steps as follows. First, set a length shorter than excess time (Te) according to the prescribed margin time (Tm). Second, set length in the same way as the excess time. Third, set length longer than the excess time (Te) by the prescribed margin time (Tm). Finally, the result can be determined in a manner similar to that for the current time synchronization period (or default time synchronization period). The example in Figure 7, the length of the next synchronization period time (Pn+1) can be determined the same or longer than the current time synchronization period (Pn) on the basis of the excess time (Te). The length of the next time synchronization period (Pn+1) can be determined into as follows: First, set length shorter than excess time (Te) by prescribed margin time (Tm). Second, set length in the same way as the excess time. Third, set length longer than excess time (Te) by a prescribed margin time (Tm). Finally, the result can be determined in a manner similar to that for the current time synchronization period.

An in-car network is a network in which all ECUs are statically configured and do not change in their life-times. So, the node with the best clock inside of the in-car network is manually set. The method described above is used in the in-car network to determine the optimal synchronization message period, thereby to minimizing traffic, with the nodes within the in-vehicle network, including the ECUs, performing optimal synchronization.

VI. EXPERIMENT

A. *Experiment construction*

The time synchronization experiments for IEEE 802.1AS were performed using a switch that can support Ethernet AVB with a number of embedded systems consisting of one master and multiple slaves for Ethernet AVB. The default value of the synchronization interval on IEEE 802.1AS was set to 0.125s ~ 1s. Time synchronization for the nodes was tested as shown in Figure 9. After performing the synchronization algorithm between the master and multiple slaves, each of the nodes generated a digital signal at the same time.

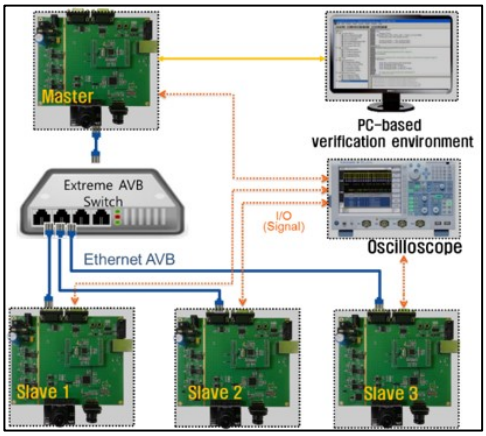

Fig 8. Experimental setting used to measure the performance of IEEE 802.1as time synchronization in Ethernet AVB Networks

B. *Experiment results*

The results for the performance of the time synchronization experiment were less than about 1μs (see Fig. 9). The performance results were satisfactory, with synchronization accuracy similar to that of current IEEE 802.1AS standards. Each synchronization message period was set to 0.125s, 0.5s and 1s. The synchronization algorithm, including offset correction and rate correction, was performed in each synchronization message period. Experimental results show that synchronization performance depends on sync message period. By effectively controlling the sync message period, synchronization performance can be maintained while minimizing the load on the network. By setting a relatively short period in the initial sync message, the clock synchronization between the master and multiple slaves was able to be quickly performed. Through appropriate adjustment of the sync message period, goal synchronization performance

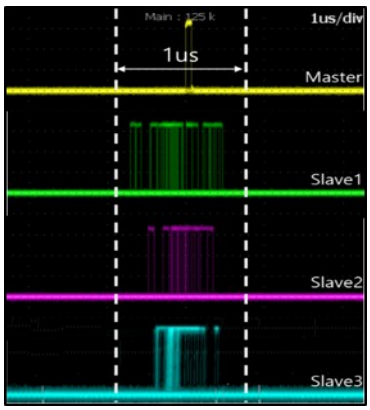

Fig 9. Experimental results from measuring the performance of IEEE 802.1AS time synchronization in Ethernet AVB Networks

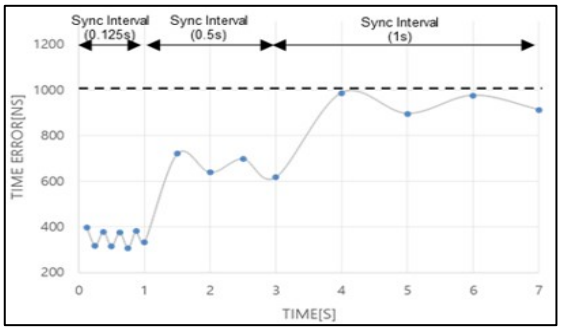

Fig 10. Experimental results from measuring the synchronization performance by adjusting the synchronization message period

(the goal synchronization performance in in-vehicle network is within 1μs [4]) using minimal sync messages can be maintained.

VII. CONCLUSION

IEEE 802.1AS time synchronization was studied for use in actual Ethernet AVB networks rather than in a simulated environment. The IEEE 802.1AS time synchronization performance was found to have an accuracy of less than 1μs. Recently, bandwidth use in in-vehicle networks has rapidly increased due to the need for interconnecting a variety of consumer electronic devices, infotainment terminals, and multimedia applications. Thus, devices that require high-precision within 1μs can be implemented using the IEEE 802.1AS protocol in Ethernet AVB. The optimal synchronization message period is automatically determined to minimize traffic through the use of the method proposed above, and the nodes inside the in-vehicle network, including the ECUs, can operate with the best synchronization performance possible.

ACKNOWLEDGMENT

This research was supported by Basic Science Research Program through NRF of Korea, funded by MOE(NRF-2010-0020210)

REFERENCES

[1] Navet, Nicolas, et al. "Trends in automotive communication systems." Proceedings of the IEEE 93.6 (2005): 1204-1223.
[2] T. Nolte, H. Hansson, and L. L. Bello, “Automotive Communications – Past, Current and Future,” in Proceedings of 10th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA’05), Sep. 2005, pp. 985–992 (volume 1).
[3] R. Freymann, “Anforderungen an das Automobil der Zukunft,” The 2nd Mobility Forum, Munich. [Online]. Available: http://www.munichnetwork.com/fileadmin/userupload/ konferenzen/ mobilitaetsforum-2/071128MUN Prof Freymann Raymond.pdf
[4] Lim, Hyung-Taek, et al. "IEEE 802.1 AS time synchronization in a switched Ethernet based in-car network." Vehicular Networking Conference (VNC), 2011 IEEE. IEEE, 2011.
[5] R. Bruckmeier, “Ethernet for Automotive Applications,” 2010, freescale Technology Forum, Orlando. [Online]. Available: http://www.freescale.com/files/ftf 2010/Americas/WBNR FTF10 AUT F0558.pdf
[6] S. Tuohy, M. Glavin, E. Jones, M. Trivedi and L. Kilmartin. "Next generation wired intra-vehicle networks, a review."Intelligent Vehicles Symposium (IV), 2013 IEEE. IEEE, 2013
[7] http://ieee802.org/1/pages/avbridges.html
[8] Kim, Yong Ju, et al. "Performance of IEEE 802.1 AS for automotive system using hardware timestamp." Consumer Electronics (ISCE 2014), The 18th IEEE International Symposium on. IEEE, 2014.
[9] Road Vehicles—Controller Area Network (CAN)—Part 4: Time-Triggered Communication, ISO 11 898-4, 2000.
[10] J. Ferreira, P. Pedreiras, L. Almeida, and J. A. Fonseca, “The FTT-CAN protocol for flexibility in safety-critical systems,” IEEE Micro (Special Issue on Critical Embedded Automotive Networks), vol. 22, no. 4, pp. 46–55, July–Aug. 2002.
[11] FlexRay Consortium. (2004, Jun.) FlexRay Communication System, Protocol Specification, Version 2.0. [Online]. Available: http://www.flexray.com
[12] D. Millinger and R. Nossal, The Industrial Communication Technology Handbook, R. Zurawski, Ed. Boca Raton, FL: CRC, 2005.
[13] Garner, Geoffrey M., Aaron Gelter, and Michael Johas Teener. "New simulation and test results for ieee 802.1 as timing performance." Precision Clock Synchronization for Measurement, Control and Communication, 2009. ISPCS 2009. International Symposium on. IEEE, 2009.
[14] Heidinger, Emanuel, et al. "A performance study of Audio Video Bridging in aeronautic Ethernet networks." Industrial Embedded Systems (SIES), 2012 7th IEEE International Symposium on. IEEE, 2012.
[15] Geoffrey M. Garner, Description of ResE Video Applications and Requirements, presentation at May, 2005 IEEE 802.3 ResE SG meeting,Austin, TX, USA, May 16, 2005.
[16] Geoffrey M. Garner, Description of ResE Audio Applications and Requirements, presentation at May, 2005 IEEE 802.3 ResE SG meeting, Austin, TX, USA, May 16, 2005.
[17] Geoffrey M. Garner, End-to-End Jitter and Wander Requirements for ResE Applications, presentation at May, 2005 IEEE 802.3 ResE SG meeting, Austin, TX,USA, May 16, 2005.
[18] A. Kern, H. Zinner, T. Streichert, J. N¨obauer, and J. Teich. Accuracy of ethernet avb time synchronization under varying temperature conditions for automotive networks. In Proceedings of the 48th Design Automation Conference, pages 597–602. ACM, 2011.
[19] J. Imtiaz, J. Jasperneite, and L. Han. A performance study of Ethernet Audio Video Bridging (AVB) for Industrial real-time communication. In Emerging Technologies & Factory Automation, 2009. ETFA 2009. IEEE Conference on, pages 1–8. IEEE, 2009.
[20] Audio Video Bridging, Wikipedia, http://en.wikipedia.org/wiki/Audio_Video_Bridging
[21] IEEE Std 802.1AS, Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks, 2011.
[22] IEEE Std 1722, IEEE Standard for Layer 2 Transport Protocol for Time Sensitive Applications in Bridged Local Area Networks, 2011.
[23] IEEE Std 802.1Qat, IEEE Standard for Local and metropolitan area networks, Virtual Bridged Local Area Networks, Amendment 14: tream Reservation Protocol (SRP), 2010.
[24] IEEE Std. 802.1Qav, IEEE Standard for Local and metropolitan area networks, Virtual Bridged Local Area Networks, Amendment 12: Forwarding and Queuing Enhancements for Time-Sensitive Streams, 2009.
[25] http://www.ieee802.org/1/pages/802.1ba.html
[26] “IEEE Draft Standard for Local and Metropolitan Area Networks – Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks (IEEE P802.1AS/D7.7),” Nov. 2010.
[27] “IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems (IEEE 1588-2008),” Jul.2008.
[28] IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, IEEE Std 1588TM–2008,24 July 2008.
[29] D. L. Mills, “Improved algorithms for synchronizing computer network clocks,” IEEE/ACM Transactions on Networking, vol. 3, no. 3, pp. 245 – 254, Jun. 1995. [Online]. Available: http://dx.doi.org/10.1109/90.392384