A Case Study: Application of Network Clock Model to Heterogeneous Sensor Networks

Soyoung Hwang
Department of Software, Catholic University of Pusan, Busan 46252, Republic of Korea
Email: soyoung@cup.ac.kr

Abstract —A sensor network consists of sensor nodes that have sensing, computation and wireless communication capabilities. There are various sensor network applications, including disaster relief operations, environmental monitoring and control, precision agriculture, intelligent home and buildings, facility management, machine surveillance and preventive maintenance, medicine and healthcare, logistics, telematics, and others. Many of these applications share some basic functionalities, namely event detection, periodic measurement and tracking, all of which entail collecting and forwarding of event data. For the purposes of triggering an action, it is necessary to know when an event occurs. Certainly time is an essential factor in many sensor network applications. This paper proposes, based on a cast study, a network clock model application to heterogeneous sensor networks. The network clock model is a scheme by which a consistent notion of time can be shared among sensor networks' devices in order to enable the combination of time and data for applications. This work serves to demonstrate the acquisition, keeping and distribution of standard time in a heterogeneous sensor network platform based on the network clock model.

Index Terms—Sensor networks, network clock model, time representation, time synchronization

I. Introduction

A sensor network comprises to a group of spatially spread sensor nodes that have sensing, computation and wireless communication capabilities. Sensor nodes monitor and record the physical conditions of an environment, and their collected data are organized at a central location.

A sensor network's application area is very extensive. One of the most notable application types is disaster relief operations. Others are precision agriculture, intelligent home and buildings, facility management, machine surveillance and preventive maintenance, medicine and healthcare, logistics, telematics, and so on [1-2].

Many of these applications share some basic functionalities such as event detection, periodic measurement and tracking, all of which entail collecting and forwarding of event data [3]. For the purposes of triggering an action in such applications, it is necessary to know when an event occurs. Indeed, time is an essential factor in many sensor network applications.

A scheme for combining time and data, in sensor networks, has emerged as a basic requirement. Therefore, the author has proposed a network clock model that allows for sharing of a consistent notion of time among devices in sensor networks and the Internet of Things [4], [5].

This paper presents a case study of an application of a network clock model to heterogeneous sensor networks. This work serves to demonstrate the acquisition, keeping and distribution of standard time in a heterogeneous sensor network platform based on the network clock model.

The remainder of this paper is organized as follows. In Section 2, related work and the motivation of the present study are discussed. Section 3 describes the architecture for the network clock model application. Next, Section 4 presents the results of a case study. Finally, Section 5 concludes this paper.

II. Related Work

Time is one of the most important issues for many applications and protocols in sensor networks. Sensor network applications require the highest degrees of time synchronization for object tracking, consistent-state updates, duplicate detection, and temporal-order delivery. Additionally to such domain-specific requirements, sensor network applications, like typical distributed systems, rely often on synchronization for secure cryptography, future-action coordination, and logged-event ordering in system debugging, among other purposes [6], [7].

Several time-synchronization schemes have been proposed for sensor networks. They can be categorized according to their characteristics. The following section presents metrics that can be employed to analyze the characteristics of time synchronization in sensor networks and categorize them [8].

A. *Metrics for Categorization of Time Synchronization in Sensor Networks*

These metrics have been derived from analyses of conventional time-synchronization algorithms and time-synchronization schemes in sensor networks.

1) *Cock management model*

- **Event ordering**: This model is for ordering of occurrences of events or messages in sensor networks. It does not require processes such as

Manuscript received October 25, 2018; revised June 3, 2019.
Corresponding author email: soyoung@cup.ac.kr
doi:10.12720/jcm.14.7.587-592

maintaining of time difference between nodes or correcting of its own clock.

- **Relative clock**: This model achieves time synchronization by maintaining the time difference between sensor nodes.
- **Adjusted clock**: This model achieves time synchronization by calculating and correcting sensor-node clock errors. This can be categorized into internal synchronization or external synchronization according to the type of clock source.

2) *Structure of time-synchronization algorithm*

- **Symmetric structure**: All of the sensor nodes that participate in the time-synchronization algorithm have the same role and execution.
- **Asymmetric structure**: The roles of sensor nodes that participate in the time-synchronization algorithm are different, such as in the master-slave structure, for example.

3) *Network topology*

- **Flat topology**: The time-synchronization algorithm operates based on a flat structure, in which sensor nodes are on the same level.
- **Hierarchical topology**: The time-synchronization algorithm operates by forming a hierarchical structure among sensor nodes. Tree structures or clusters are typical.

4) *Clock source*

- **Internal source**: Time synchronization is achieved by using the clock value of one node in a sensor network.
- **External source**: A clock source for time synchronization exists outside the network. It is typically a source, such as GPS (Global Positioning System), that provides standard time.

5) *Synchronization scope*

- **Global synchronization**: It synchronizes the times of all of the sensor nodes in the network to achieve single- as well as multi-hop network synchronization.
- **Local synchronization**: It synchronizes nodes in a certain area for sharing of certain events in a network or nodes in a single-hop.

6) *Others: Synchronization techniques that are dependent on the application of sensor networks, etc.*

B. *Time Synchronization in Sensor Networks and for Internet of Things*

Various time-synchronization protocols have been proposed for sensor networks and, recently, for the IoT as well. These protocols mostly account for new constraints such as precision and accuracy requirements, large numbers of nodes in networks, and low energy consumption. For the purposes of the present study, some of that recent work is reviewed in detail below.

In heterogeneous sensor networks, a target-tracking application requires tight synchronization. Heterogeneous sensor networks consist of resource-constrained nodes as well as resource-intensive nodes equipped with high-bandwidth sensors. A methodology for accurate time synchronization necessary for fusion of audio and video data collected from heterogeneous senor nodes is discussed in [9]. Its testbed demonstrates an accuracy on the order of microseconds over a multi-hop network, and it improves tracking performance in a multi-modal tracking application.

In order to enhance time-synchronization support in wireless sensor networks, three algorithms, namely self-correction [10], clock-prediction and analytical-correction, are proposed in [11]. These make the sensor nodes' clocks converge as quickly as possible while maintaining low energy consumption.

PSync is a visible-light-based time-synchronization protocol [12]. It utilizes an LED (Light Emitting Diode) light source to synchronize nearby devices in consideration of energy efficiency. The power efficiency in the receiver is derived from the facts that the light sensors in the receivers consume very little power and that the sampling of light sensors can be done at a high rate. Therefore, the receiver can sleep most of the time and wake up only at short intervals to perform synchronization, because the energy consumed by synchronization protocols depends on the cost of radio transmission and reception.

CoAP (Constrained Application Protocol) is the IETF (Internet Engineering Task Force) standard for sensor networks. A time-synchronization technique is proposed for CoAP-based home automation system networks [13]. It utilizes the CoAP option field and a shim header to include time-stamps; therefore, it can be applied to both IP-based and non-IP-based home automation systems.

A clock-synchronization approach has been proposed for synchronization of clocks between IoTs and Cloud [14], [15]. Based on PTP (Precision Time Protocol), it focuses on minimizing clock drift and master-slave delay.

Temperature-resilient time synchronization that is suitable for the IoT is proposed in [16]. Networks deployed in real-world conditions must cope with changes of environmental temperature that are dynamic and, so, unpredictable. Such changes can affect network nodes' clock rate and lead to clock de-synchronization that is faster than under stable-temperature device-operation conditions. Thus, temperature-resilient time synchronization was devised as a method for autonomous compensation of temperature-dependent clock-rate changes. After the calibration stage, nodes perform temperature measurements continuously in order to compensate for clock drift at run time.

An architecture for IoT-device clock synchronization is presented in [17]. Clock drift on two standard IoT platforms was examined, and the synchronization system architecture was developed to improve clock accuracy and scalability for support of large numbers of clients.

In order for such time-synchronization protocols to be effectively utilized, a clock model allowing for sharing of a consistent notion of time among network devices is required. In previous studies, the author has proposed a network clock model that allows for sharing of a consistent notion of time among devices in sensor networks and the IoT [4-5]. This paper considers a case study of a network-clock-model application to heterogeneous sensor networks.

III. Architecture for Network Clock Model Application

A. Network Clock Model

In order to maintain current standard time, a 64-bit system time structure was defined for sensor nodes and IoT devices. The first 32-bits contains the value in seconds since 00:00:00 January 1900 UTC. UTC (Universal Time Coordinated) is the world's primary time standard. The second 32-bits contain the value in micro-seconds.

In monitoring and controlling the operations of sensor network applications, standard time is essential in order to trace event occurrence points and analyze data. Additionally, the relative strength of latency and accuracy are improved if the time information is recorded in the device where the event originated. If all of the devices in the networks share a consistent notion of time, building efficient and robust collaborative services is easier.

For the purposes of accessing the system time structure and providing date and time services, the following basic functions are employed: settimeofday sets the device's system time structure to the given value; gettimeofday provides the current time of the device in the system time structure format; date provides the Gregorian calendar date and current time of the device [5].

B. Architecture for Application

For application of a network clock model to heterogeneous sensor networks, an architecture is defined as shown in Fig. 1.

Fig. 1. Architecture for network clock model application

The network clock model maintains current standard time for sensor nodes. As such, needs a scheme for acquisition, keeping and distribution of standard time.

Additionally, for application to heterogeneous sensor networks, three standard sensor network prototyping platforms are considered: Aduino, Raspberry Pi, and ZigBee sensor node. Fig. 2 presents the architecture for network clock model application utilizing the prototyping platforms. Also, it should be noted that the sensor network discussed herein is one without any infrastructural connection such as the Internet.

Arduino is an open-source electronic prototyping platform enabling users to create interactive electronic objects. The Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote teaching of basic computer science in developing countries. ZigBee is an IEEE 802.15.4-based specification for a suite of high-level communication protocols used to create personal area networks with small, low-power digital radios.

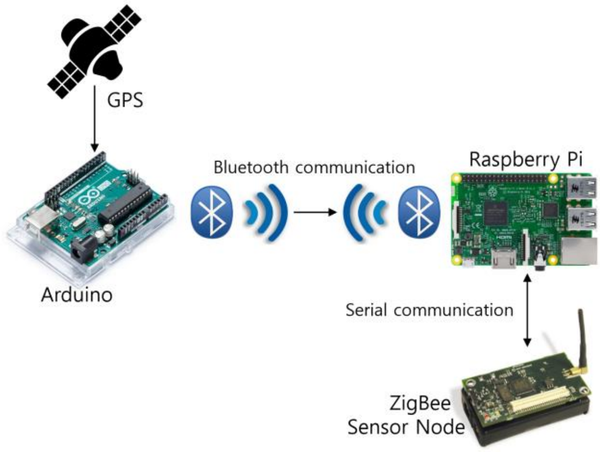

Fig. 2. Architecture for network clock model application utilizing prototyping platforms

Arduino uses GPS as a reference clock to acquire current standard time. GPS is a global navigation satellite system owned by the United States government. It provides Precise Positioning Service (PPS) and Standard Positioning Service (SPS) as its two primary services. PPS, an encrypted service, is intended exclusively for military or other, authorized governmental users. SPS, which is available free of charge, is used by hundreds of millions of users globally, both civil and commercial. PPS and SPS provide navigation signals that enable users' receivers to determine position, velocity and UTC with reference to the U.S. Naval Observatory (USNO) [18].

Arduino keeps standard time based on the network clock model and its timer. It distributes current standard time to Raspberry Pi via Bluetooth communication. It also can distribute standard time to other sensor nodes.

Raspberry Pi has a Linux-based operating system. It has its own system time and application programming interfaces (APIs) for system time. Therefore, Arduino provides standard time in the format of the system time in Raspberry Pi, and Raspberry Pi keeps standard time based on its own system time.

Raspberry Pi distributes current standard time to ZigBee sensor node via serial communication in the format of the network clock model. It also can distribute standard time to other sensor nodes via various communication protocols.

ZigBee sensor node acquires standard time through Raspberry Pi and keeps standard time based on the network clock model and its timer. It can distribute current standard time to other ZigBee sensor nodes via ZigBee communication.

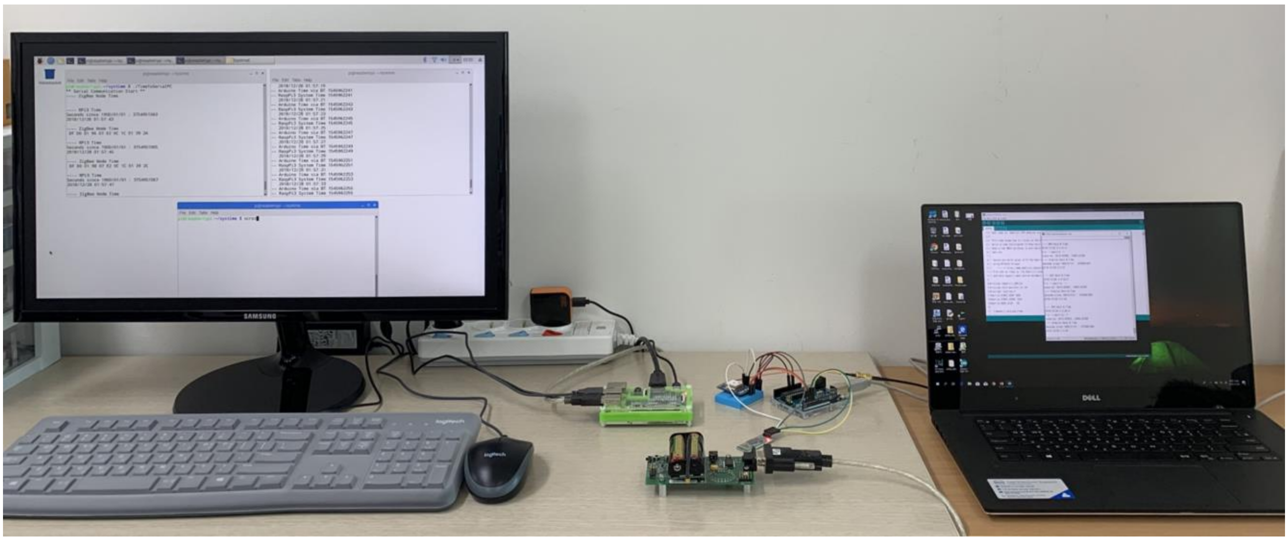

Fig. 3. Experimental environment

IV. Case Study

The experimental environment of the case study is illustrated in Fig. 3. Fig. 4 presented the elements utilized in the experiment.

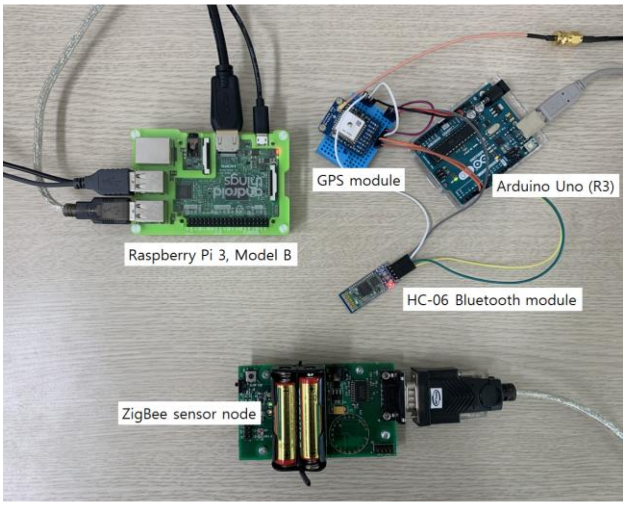

Fig. 4. Elements utilized in experiment

Arduino Uno (R3) is a microcontroller board based on the ATmega328P. The GPS module is Adafruit Ultimate GPS version 3. The Bluetooth module is the HC-06 wireless Bluetooth transceiver for Arduino. Raspberry Pi 3 Model B has a 64-bit quad core processor, on-board WiFi, Bluetooth and USB boot capabilities. ZigBee sensor node is based on the Atmel ATmega128L, a low-power microcontroller with an IEEE 802.15.4 radio.

As described in Section 3, each of the nodes acquires, keeps and distributes current standard time. In order to verify the proper operation of each nodes, Arduino Uno transmits its system time and GPS time to the monitoring laptop computer every 2 seconds. Raspberry Pi 3 prints out its system time and Arduino's system time that is received via Bluetooth communication to the terminal every 2 seconds. ZigBee sensor node transmits its system time and Raspberry Pi 3's system time that is received via serial communication to Raspberry Pi 3's terminal every 2 seconds.

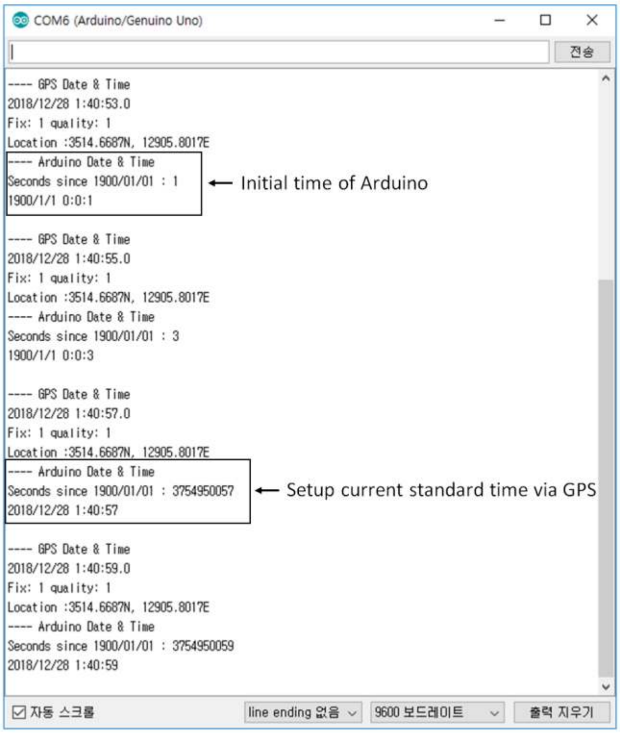

Fig. 5. Acquisition of standard time in Arduino Uno

Fig. 5 shows the serial monitor results for Arduino Uno (R3). When the system is started, its system time is initiated. The system time indicates the time in units of seconds and micro-seconds since 00:00:00 January 1900 UTC. The Arduino system time of 0 seconds, therefore, corresponds to a calendar time of 00:00:00 January 1900 UTC. As can be seen in Fig. 5, the Arduino system time starts from 0. Arduino utilizes GPS as a reference clock to acquire the current standard time. In the implementation, the system time is updated to the current standard time once the GPS signal is stabilized.

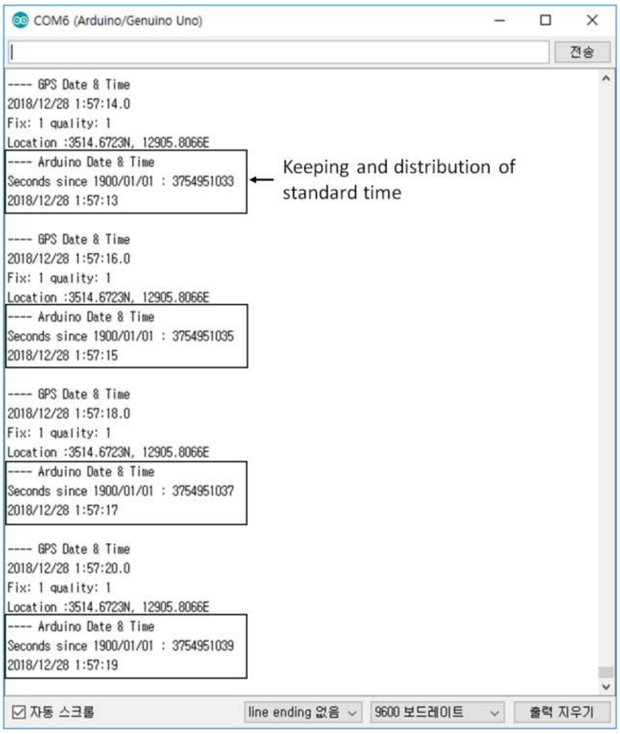

Fig. 6. Keeping and distribution of standard time in Arduino Uno

Fig. 6 shows that Arduino Uno keeps and distributes the current standard time. When the system time is updated to the current standard time, it maintains that time based on the network clock model and its timer. As can be seen, it can acquire, keep and serve standard time for an application.

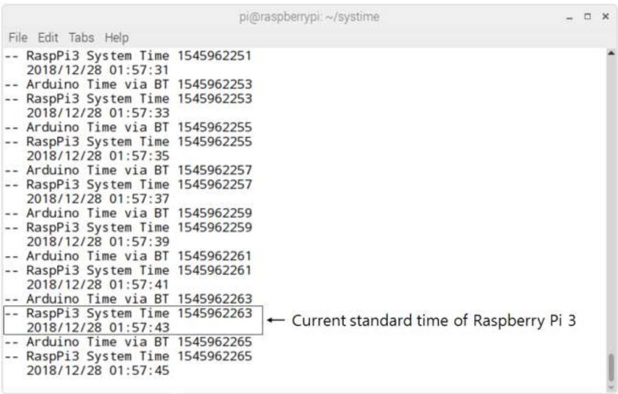

Fig. 7. Acquisition and keeping of standard time in Raspberry Pi 3

Fig. 7 shows the terminal results for Raspberry Pi 3. Raspberry Pi 3 receives the standard time from Arduino Uno via Bluetooth communication, and updates its system time to the current standard time. Arduino provides the standard time in the format of the system time in Raspberry Pi 3, because Raspberry Pi 3 has a Linux-based operating system and its own system time.

When the system time of Raspberry Pi 3 is updated, it maintains the standard time based on its own system time. As can be seen, it too can acquire, keep and serve standard time for an application.

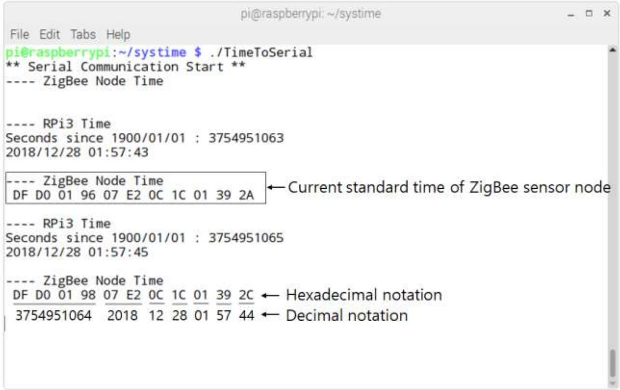

Fig. 8. Acquisition and keeping of standard time in ZigBee sensor node

Fig. 8 shows the terminal results for the ZigBee sensor node. Because there is a delay in showing the ZigBee sensor node's time, it is behind the Raspberry Pi 3 time in the result.

The ZigBee sensor node receives the standard time from Raspberry Pi 3 via serial communication, and updates its system time to the current standard time. Raspberry Pi 3 provides the standard time in the format of a network clock model. When the system time is updated, the ZigBee sensor node keeps the standard time based on the network clock model and its timer. And, as can be seen in the Fig. 8 result, it also can acquire, keep and serve standard time for an application.

V. Conclusions

A sensor network consists of sensor nodes that have sensing, computation and wireless communication capabilities. There are various applications of a sensor network, such as disaster relief operations, environmental monitoring and control, precision agriculture, intelligent home and buildings, facility management, machine surveillance and preventive maintenance, medicine and healthcare, logistics, telematics, and so on.

Many of these applications share some basic functionalities, namely event detection, periodic measurement, and tracking, all of which entail collecting and forwarding of event data. For the purposes of triggering an action, it is necessary to know when an event occurs. Certainly, time is an essential factor in many sensor network applications.

In the author's previous work, a network-clock-model-based scheme for combining time and data for sensor networks and the IoT was proposed. It allows for sharing of a consistent notion of time among devices in sensor networks and the IoT.

This paper discusses a case study of an application of a network clock model to heterogeneous sensor networks. In the study, an architecture for application was defined, and experimentation with three prototyping platforms without connection to any infrastructure such as the Internet was performed. The results serve to demonstrate the acquisition, keeping and distribution of standard time in the heterogeneous sensor network platform based on the network clock model.

As for future work, a new time-management scheme that can meet the requirements of accuracy and precision is required. For accuracy, a new time-synchronization scheme is needed, and for precision, research on a clock source and handling mechanism is necessary.

ACKNOWLEDGMENT

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2017R1A2B4009 167).

REFERENCES

[1] H. Karl and A. Willig, *Protocols and Architectures for Wireless Sensor Networks*, West Sussex, England: John Wiley & Sons Ltd., 2005.

[2] S. Hwang and D. Yu, "Remote monitoring and controlling system based on ZigBee networks," *International Journal of Software Engineering and its Applications*, SERSC, vol. 6, no. 3, pp. 35–42, July 2012.

[3] K. Römer, P. Blum, and L. Meier, "Time synchronization and calibration in wireless sensor networks," in *Handbook of Sensor Networks: Algorithms and Architectures*, Ivan Stojmenovic (Ed.), West Sussex, England: John Wiley & Sons Ltd., 2005.

[4] S. Hwang and S. S. Joo, "Global Time Service in Wireless Sensor Networks," in *Proc. 9th International Symposium on Communication and Information Technology*, Incheon, South Korea, September 2009, pp. 382–383.

[5] S. Hwang, "A network clock model for time awareness in the Internet of things and artificial intelligence applications," *Journal of Supercomputing*, pp. 1–20, February 2019.

[6] J. Elson and K. Römer, "Wireless Sensor Networks: A new regime for time synchronization," *ACM Computer Communication Review*, vol. 33, no. 1, pp. 149–154, 2003.

[7] D. L. Mills, *Computer Network Time Synchronization*, 2nd ed. Boca Raton, FL.: CRC Press, 2011, ch. 1, pp. 1–14.

[8] S. Hwang, "Reliable time synchronization in wireless sensor networks," Ph.D. dissertation, Department of Computer Science, Pusan National University, Busan, Republic of Korea, February 2006.

[9] I. Amundson, M. Kushwaha, B. Kusy, P. Volgyesi, G. Simon, X. Koutsoukos, and A. Ledeczi. Time Synchronization for Multi-Modal Target Tracking in Heterogeneous Sensor Networks. [Online]. Available: http://www.isis.vanderbilt.edu/sites/default/files/KalamataWorkshop07.pdf

[10] L. T. Bruscato, E. P. Freitas, and T. Heimfarth, "Self-Correcting time synchronization support for wireless sensor networks targeting applications on internet of things," *International Federation of Automatic Control*, vol. 49, no. 30, pp. 355–360, 2016.

[11] L. T. Bruscato, T. Heimfarth, and E. P. de Freitas, "Enhancing time synchronization support in wireless sensor networks," *Sensors*, vol. 17, no. 12, pp. 2956, December 2017.

[12] X. Guo, M. Mohammad, S. Saha, M. C. Chan, S. Gilbert, and D. Leong, "PSync: Visible light-based time synchronization for internet of things (IoT)," in *Proc. 35th Annual IEEE International Conference on Computer Communications*, San Francisco, CA, USA, 10-14 April 2016.

[13] S. Son, N. Kim, B. Lee, C. H. Cho, and J. W. Chong, "A time synchronization technique for COAP-based home automation systems," *IEEE Transactions on Consumer Electronics*, vol. 62, no. 1, pp. 10–16, February 2016.

[14] R. S. Jha and P. Gupta, "Clock synchronization in IoTs network through cloud," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 12, no. 1, pp. 337–350, 2017.

[15] R. S. Jha and P. Gupta, "Clock synchronization in IoT network using cloud computing," *Wireless Personal Communications*, vol. 97, no. 4, pp. 6469–6481, December 2017.

[16] A. Elsts, X. Fafoutis, S. Duquennoy, G. Oikonomou, R. Piechocki, and I. Craddock, "Temperature-Resilient time synchronization for the internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 5, pp. 2241–2250, May 2018.

[17] S. K. Mani, R. Durairajan, P. Barford, and J. Sommers. "An architecture for IoT clock synchronization," in *Proc. 8th ACM International Conference on Internet of Things*, Santa Barbara, CA, October, 2018.

[18] E. Kaplan and C. J. Hegarty, *Understanding GPS/GNSS: Principles and Applications*, 3rd Ed. Boston, London, Artech House, 2017.

Soyoung Hwang received her B.S., M.S., and Ph.D. degrees in Computer Science from Pusan National University, Busan, Korea, in 1999, 2001, and 2006, respectively. From 2006 to 2010, she was a senior researcher at the Electronics and Telecommunication Research Institute (ETRI), Daejeon, Korea. Since 2010, she has been a professor of the Department of Software at Catholic University of Pusan, Busan, Korea. Her research interests include embedded systems, sensor networks and the Internet of Things.