A Time Synchronization Technique for CoAP-based Home Automation Systems

Seung-Chul Son, Nak-Woo Kim, Byung-Tak Lee, Chae Ho Cho, and Jo Woon Chong

Abstract — *With the advent of internet-of-things (IoT)-based home automation systems, time synchronization techniques for low power sensor modules are in high demand. The constrained application protocol (CoAP) was recently standardized for sensor networks by IETF and is becoming widely adopted for home automation systems by ETSI, OMA, and oneM2M. The network time protocol (NTP) is not applicable to home automation systems due to its limited computing resources. This paper proposes a lightweight time synchronization algorithm for CoAP-based home automation system networks. The CoAP option field and a shim header are used to include time-stamps in the home automation system. The proposed scheme can thus be applied to both IP-based and non-IP-based home automation systems. In experiments with several household devices having non-IP communication interfaces, experimental results show that the proposed technique gives an average error of 1 ms and a network overhead reduction of 17% when compared to the ideal NTP service[1].*

I Index Terms — home automation, smart home, time synchronization, CoAP

I. INTRODUCTION

As user demands for communications between the home and the outside world increase, the requirement for internet-of-things (IoT) technologies for home automation systems have also increased. As a result, IoT technologies [1] have accelerated the development of communication protocols as well as sensors for home automation systems. Currently, home automation system platforms, which collect data from the sensors and appliances using commercial off-the-shelf (COTS) wireless technologies, have been developed along with the IoT technologies [2], [3]. The constrained application protocol (CoAP) was proposed to needs that are not satisfied by COTS technologies, for communications in home automation systems [4]. In addition, home applications usually require accurate and concurrent timing information in order to transmit their data in a timely manner. For example, fire alarms and intrusion warnings in home automation systems need their data to be transmitted as quickly and reliably as possible [6]. Hence, time synchronization between nodes is so crucial that it can severely affect the performance of the home automation systems.

There have been a number of studies investigating time synchronization. Time synchronization is usually performed by a sequence of ordering sensor data obtained from sensor nodes, distinguishing duplicate events occurring from peripheral sensor nodes, and publishing a sensor node's timestamp and concurrency control [5]. Since sensor nodes in the home automation system alternate between sleep and active modes to be more efficient at power consumption (Fig. 1), accurate time scheduling can play an important role in saving energy costs by considering the time-varying price of electric power in a home energy management system (HEMS) [7]–[9]. Specifically, global time synchronization is necessary in order to concurrently meet delay-intolerant and energy-efficient requirements. The network time protocol (NTP) was proposed in order to provide time synchronization for a larger and static network such as the internet. However, NTP is based on a client-server model in which the master has a synchronization accuracy of 1 μs through GPS, and clients are sequentially synchronized with the master. NTP enables a larger internet, with an accuracy in the tens of milliseconds while providing a smaller network with an accuracy of less than 1 ms [10]. Therefore, NTP cannot be directly applied to a sensor network.

Other time synchronization schemes, such as the timing-sync protocol for sensor networks (TPSN) [11], reference broadcasting synchronization (RBS) [12], flooding time synchronization protocol (FTSP) [13], and the IEEE 1588 protocol [14] have also been proposed in attempts to minimize resources and maximize the efficiency of sensor network communications. In TPSN, the sensor network is configured into a hierarchical topology, like a tree structure, and then a time synchronization process progresses hierarchically from the lower level to the upper level. The time synchronization is fast, though it has difficulty to respond to changes in the topology when a sensor node is inserted or removed. In RBS, a designated node such as a proxy or a remote terminal unit (RTU) broadcasts a time reference message, and nodes are then synchronized according to the relative time differences calculated by deviation between the sensor nodes.

[1] This work was supported by the ETRI R&D Program (Research project: "Optical Convergence Technology for Regional Optical Components Enhancement", 15ZC2300) funded by the Government of Korea.

S. Son is with the IT Management Device Service Team, Electronics and Telecommunications Research Institute, Gwangju, Korea (e-mail: sson@etri.re.kr).

N. Kim is with the IT Management Device Service Team, Electronics and Telecommunications Research Institute, Gwangju, Korea (e-mail: nwkim@etri.re.kr).

B. Lee is with the IT Management Device Service Team, Electronics and Telecommunications Research Institute, Gwangju, Korea (e-mail: bytelee@etri.re.kr).

C. H. Cho is with the M2Soft Co., Ltd, Seoul, Korea (e-mail: backjoe@gmail.com).

J. W. Chong is with the Worcester Polytechnic Institute, Worcester, USA (e-mail: jw9607@gmail.com).

Contributed Paper
Manuscript received 01/06/16
Current version published 03/30/16
Electronic version published 03/30/16

In a small-scale network, the transmission and access time are considerably reduced. However, as the number of sensor nodes increases, the number of messages that are transmitted also greatly increase, resulting in a large amount of power being consumed. FTSP conducts time synchronizations by flooding a global reference message. The sensor nodes revise the time error and synchronize time, depending on the error factors, through a linear regression analysis of the duplicate messages received. This method is dependent on the system environment and requires a time delay for the initial time synchronization. The precision time protocol (PTP) from the IEEE 1588 standard can be implemented on IP and UDP networks, and it is similar to NTP. PTP is generally more accurate than NTP, though it uses more messages than NTP, and the size of its message structure is of over 60 bytes. Thus, it is difficult to apply PTP to a sensor network environment [15]. Hence, more efficient time synchronization methods for sensor networks remain in high demand.

CoAP, highlighted as a technology for use in home automation systems, is an application-level protocol based on the low level standard for IEEE 802.15.4 sensor networks. In addition, CoAP is a web-based protocol that supports resource discovery based on the representational state transfer (REST) architecture. Since sensors in M2M usually make use of low-cost and low-performance processors, it is difficult to use the existing TCP/HTTP standard, though it is possible to easily convert and interact with the existing HTTP web protocol through a proxy through CoAP's RESTful feature. CoAP is primarily intended for use in transmitting sensor data, supporting multicast and asynchronous transactions in a limited network environment. In 2013, CoAP was adopted as a standard by the constrained RESTful environment (CoRE) working group of the internet engineering task force (IETF) [16]. Sensor systems and platforms have since started to support CoAP. However, time synchronization issues remain unsolved. With the aforementioned TPSN, RBS, and IEEE 1588, CoAP results in an inevitable resource waste. Therefore, a time synchronization module for CoAP is required.

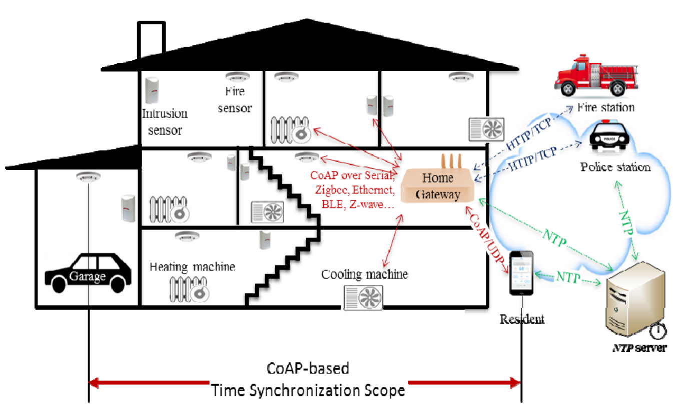


Fig. 1. Network configuration for CoAP-based home automation system.

In this paper, a novel time synchronization technique for CoAP is presented. It minimizes resource usage and network overhead to attain a high accuracy for time synchronization. A sensor network using the CoAP standard is developed to deliver home sensor data between household devices and the home gateway (Fig. 1). In particular, the proposed synchronization scheme aims at minimizing both resource usage and network overhead on the sensor nodes and gateway to achieve high time accuracy when compared to existing schemes. As far as we know, this paper is the first report about the CoAP based time synchronization technique with a practical application in the home automation system.

The remainder of this paper is organized as follows. Section II gives a detailed description of the proposed CoAP-based time synchronization technique, and Section III presents the experimental results. Finally, Section IV concludes this paper.

II. Time Synchronization Technique for CoAP-Based Home Automation System

In this chapter, a time synchronization technique is proposed for a CoAP-based home automation system. Many physical interfaces have been proposed for configuring home automation systems, including serial communication, Ethernet, BLE, Z-wave, Zigbee, WiFi, etc., which can be classified as IP-based and non-IP-based interfaces. Accordingly, this paper proposes a novel time synchronization technique for CoAP that can be utilized in both non-IP and IP environments. The proposed technique starts with a preprocessing step that adds a shim header to the CoAP protocol. This preprocessing step is used to implement the CoAP-based time synchronization technique in a non-IP environment.

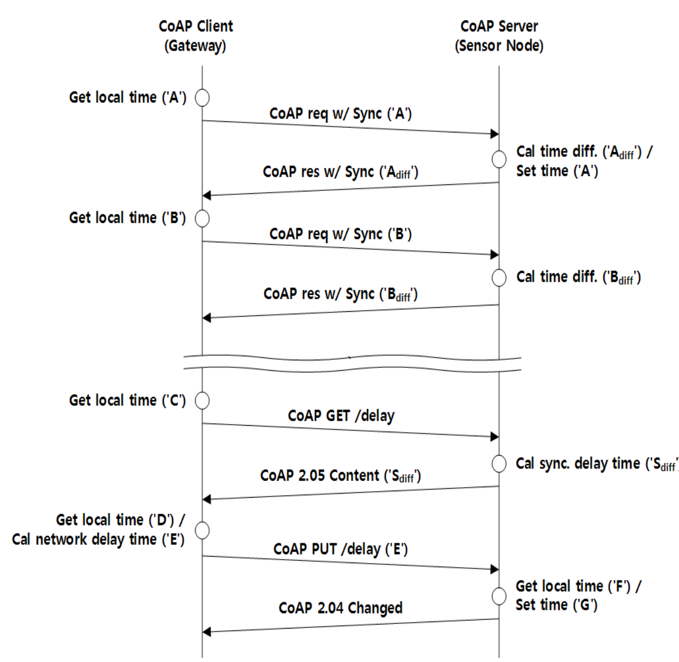


Fig. 2. CoAP-based time synchronization process.

The proposed scheme consists of two steps: a *time-sharing process* and a *network delay compensation process*. The time-sharing process is a procedure for updating the time information of the CoAP servers (e.g., household device) by obtaining the time information of the CoAP clients (e.g., home gateway, home proxy, and RTU). Several messages are exchanged until both the server and client are matched. The network delay compensation process is a procedure for fine-tuning synchronization by compensating for the network delay. The CoAP client records the current timestamp in the CoAP message when requesting the network delay calculation, and then sends it to the CoAP server. The CoAP server sends the CoAP client the timestamp difference between the timestamp recorded in the received CoAP message and the CoAP server's current timestamp. The CoAP client subsequently calculates the network delay according to the previously recorded timestamp and the current message's arrival timestamp, and sends the CoAP server the calculated network delay using the PUT method. The CoAP server compensates for this synchronization time by using the summation of the current time and the network delay.

Fig. 2 presents an example of the time synchronization process. In the time-sharing process, the transmission and response of the time information are piggybacked using the CoAP extension option. In the network delay compensation process, the CoAP client first sends a 'GET /delay' request, and the CoAP server then sends back a response message and the synchronization delay. Next, the CoAP client calculates the network delay, before sending the network delay information through a 'PUT /delay' request. Finally, the CoAP server compensates for the time by adding the network delay to the current time.

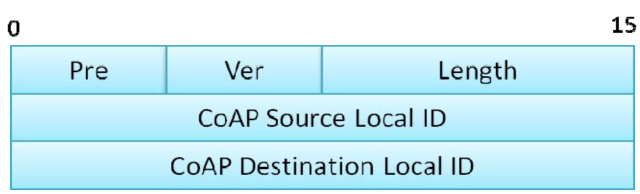


Fig. 3. Shim header format for supporting a non-IP environment, such as ZigBee, RS485, UART, etc.

A. *CoAP on the Non-IP Environment*

In this paper, the basic CoAP functionality is only implemented from among the full IETF CoAP specification. Rare functions are omitted in order to offer a simple and lightweight CoAP protocol for general use. Also, the shim header is designed and sensor identifiers are implemented for the sensor communication interfaces, to support non-IP environments like ZigBee, RS485, and UART. The shim header is inserted in front of the CoAP header, instead of an IP and UDP header, and this contains the addresses of the CoAP sender and receiver as well as the length of the CoAP header and its payload. Fig. 3 shows the basic format of the shim header. The following is an explanation about each field in the header.

- Pre: Preamble is a 4-bit field and indicates the start of the shim header: '0xa' is currently used.
- Ver: Version is also a 4-bit field, and the version number is currently '1'.
- Length: Length is the length of the CoAP header and its payload, followed by the shim header. This has a value ranging between 1 and 255.
- CoAP Source Local ID, CoAP Destination Local ID: The local IDs of the sender and receiver for the CoAP message; each field has a 16-bit field.

Fig. 4 presents an example of a basic confirmable CoAP request and the response in a piggybacked manner. In this case, the client and server are connected via a non-IP network. The client can be a data collector node or a proxy node, and the server may be a temperature sensor. In this example, it is assumed that the client has an ID of 0x0000 and the server has an ID of 0x0001.

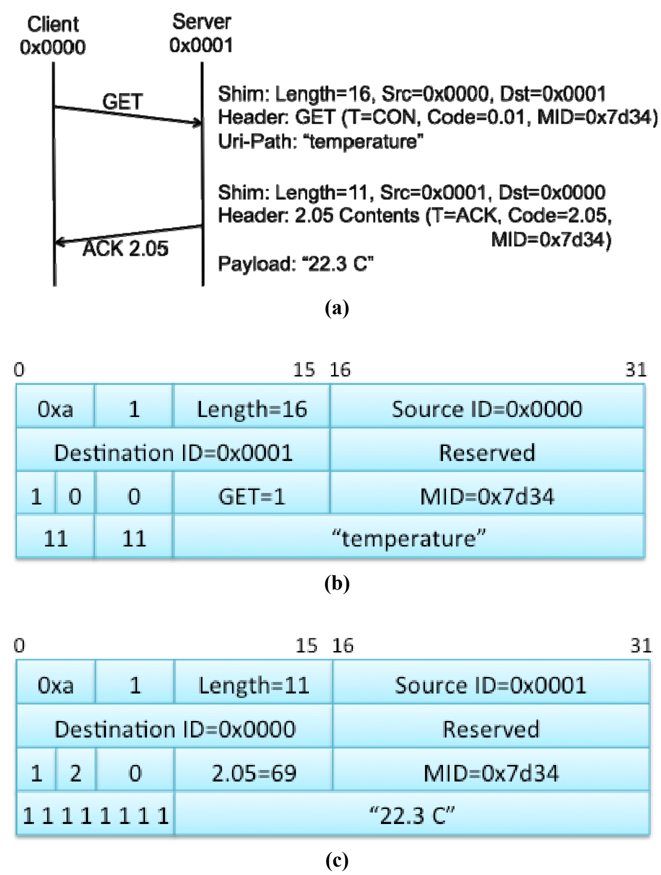


Fig. 4. Shim header usage: (a) example of a CoAP GET request and response, (b) CoAP packet for the GET request, and (c) CoAP packet for the response, including the shim header.

B. *Time-sharing process*

In the first step of the CoAP-based time synchronization technique, the CoAP server and the client share the time with each other. The reference point for this time-sharing is the CoAP client's current time. A timestamp having a 4-byte length is used with the SYN option for time synchronization. In general, a CoAP GET request is used to obtain sensor information. The CoAP client uses a piggyback method that puts the timestamp to the SYN option, as in Fig. 5(a). Note that '2500' is a message identifier that prevents duplicate transmissions. The token is a field used to confirm message pairs. Fig. 5(b) shows the CoAP server's response, including the SYN option; this value is 'A_{diff}', which is the time difference between two machines before the CoAP server's time is updated on the CoAP client (Fig. 5). In this way, if the CoAP request and response messages are repeated at least twice, the value of the SYN option in the CoAP response message that the CoAP server sends to the CoAP client is zero. This is a server-to-client time-sharing point, and it is an entry point for the network delay compensation process.

CON GET coap://coap.svr.net/timestamp [2500]
Token : ed
Sync : 'A'

(a)

ACK 2.05 Content [2500]
Token : ed
Sync : 'A_{diff}'

(b)

Fig. 5. Time-sharing process: (a) CoAP client request, and (b) CoAP server response.

C. Network Delay Compensation Process

After the time-sharing process, the algorithm progresses to the network delay compensation process through another CoAP request and CoAP response sequence. Again, the CoAP client informs the CoAP server of the start of the network delay compensation process by recording its current timestamp 'C' with the '/delay' uri-path (Fig. 6(a)). The CoAP server calculates the time difference 'S_{diff}'' between 'C' and the CoAP server's current time, and it then reports the synchronization delay to the CoAP client (Fig. 6(b)).

Fig. 7 shows the CoAP message for the network delay compensation process. The CoAP client looks for the difference in the CON message send time 'C' and the ACK response receive time 'D' and calculates the network delay, such that 'E = (D-C)/2'. When the CoAP client sends 'E' through to a PUT message, the CoAP server sets the new current time 'G' by adding 'E' to the CoAP server's current time 'F', and then responds with '2.04 Changed' to the CoAP client.

D. Time Synchronization Reconfirmation Process

The time synchronization reconfirmation step is the process that confirms that the synchronized time persisted. Similar to the network delay compensation process, the algorithm calculates the synchronization time 'S_{diff}', which is the difference of the time information 'C' from the CoAP client and the CoAP server's current time. At this point, if the synchronization time 'S_{diff}' exceeds the previously-specified allowable error, both the CoAP server and the CoAP client reset, restarting from the time-sharing process mentioned above.

CON GET coap://coap.svr.net/delay [2508]
Token : 37
Sync : 'C'

(a)

ACK 2.05 Content [2508]
Token : 37
Sync : 'S_{diff}'

(b)

Fig. 6. Calculation to synchronize the time delay: (a) CoAP client request, and (b) CoAP server response.

CON PUT coap://coap.svr.net/delay [2509]
Token : 7f
Sync : 'E'

(a)

ACK 2.05 Content [2509]
Token : 7f
Sync : 'G'

(b)

Fig. 7. Network delay compensation process: (a) CoAP client request, and (b) CoAP server response.

III. System Implementation

A. Non-IP CoAP-Based Home Automation Network

A sample implementation of the proposed time synchronization system is illustrated in Fig. 8. As depicted, a non-IP CoAP-based home automation system is designed for monitoring and controlling household devices. The system is made of a home gateway, three CoAP-server-installed home automation devices, and an analyzer to confirm the synchronized time. The home gateway and three devices are connected using Zig-Bee, RS485, and UART.

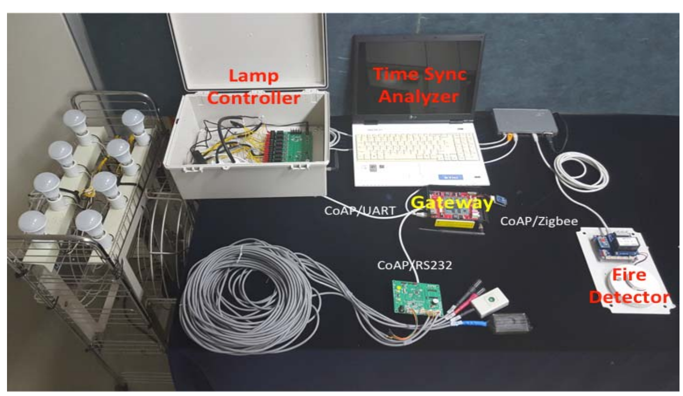


Fig. 8. System implementation for the CoAP-based time synchronization.

B. Home Gateway

A general home gateway collects information from household devices being provided to the user and relays the user's control commands toward household devices in an IP network environment. However, home gateway developed in this paper works not only in an IP network but also in non-IP networks such as UART, RS285, or ZigBee (Fig. 9). The developed gateway is synchronized with an outside NTP server and works as a time reference server for the implemented home automation systm. It includes a CoAP client that is used to communicate with the CoAP-server-installed home automation devices.

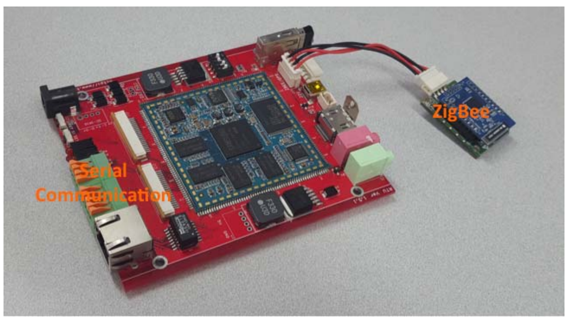


Fig. 9. Home gateway having non-IP communication interface.

C. Home Automation Devices

To demonstrate the feasibility and effectiveness of the proposed system, three devices were developed: a fire detector, lamp controller, and integrated environmental sensor. The CoAP server that supports the proposed CoAP time synchronization process is installed in each device. These devices are depicted in Fig. 10(a), (b), and (c).

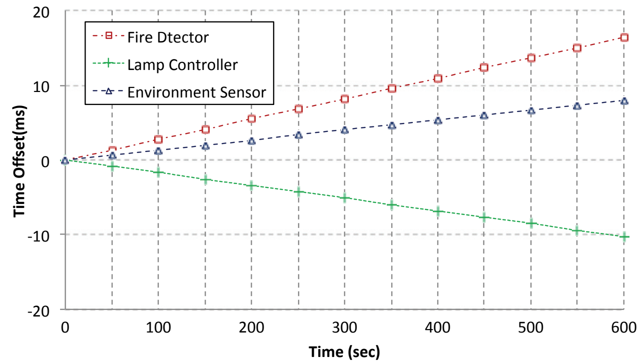
(a) (b) (c)

Fig. 10. (a) CoAP-based fire detector, (b) CoAP-based lamp controller, and (c) CoAP-based integrated environmental sensor.

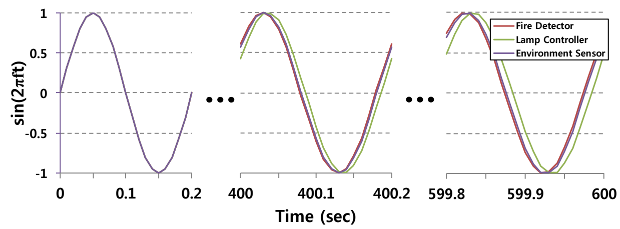
Fire Detector: A conventional fire alarm is integrated with a ZigBee microcontroller, (Fig. 10(a)). Using this prototype, users can detect fire signals. By using CoAP over ZigBee communication in conjunction with the home gateway, signals are synchronized with the outside world so that it is possible to provide accurate time information.

Lamp Controller: A prototype lamp controller was developed and integrated with a UART communication interface (Fig. 10(b)). Users can remotely monitor and control the state of the lamp. Specifically, real-time operation is possible because the accurate time information is obtained through the CoAP time synchronization from the home gateway.

Environmental Sensor: A prototype sensor incorporated four types of sensors: temperature-humidity, luminance, CO_2, and NH_3 sensors was developed and integrated with an RS485 communication interface (Fig. 10(c)). Users can remotely monitor the environmental state of the house. Real-time operation is also possible due to the accurate time information provided using the CoAP time synchronization.



(a)



(b)

Fig. 11. Analyses of the skewed clock in devices in comparison with the NTP server: (a) time offset of synchronized home gateway and non-synchronized devices, and (b) skewed shape of devices' clock when the devices report sine function values of the time parameter (frequency f=5).

IV. EVALUATION

The time synchronization performance of the implemented system was verified via a comparison with the proposed CoAP-based time synchronization technique and the ideal time using an NTP server. Fig. 11 shows the skewed clock condition of each device in comparison with the NTP server. In brief, the gateway is connected to the NTP server, and it is synchronized using a standard internet-based NTP protocol. The minimum synchronization period between the home gateway and the NTP server was set to 8 s, as specified in the NTP standard. Each device tends to linearly change at a constant rate, as a general crystal oscillator having a clock drift of 100 ppm was chosen as the clock source for these devices. An error rate of 100 ppm can generate a time error of about 9 s per day. Therefore, changes in the time offset are inevitable, as seen in Fig. 11(a). The increase in the time offset implies that the device's time is faster than that of the gateway, but the decrease in the time offset indicates the contrary.

Using a CoAP observation function, Fig. 11(b) shows that each device sent reports at up to 100 times per second. The

CoAP observation sends notification reports whenever the state of a resource changes, rather than being based on a polling method that uses a request and response message pair. The value that each device reports is the sine function value of the time parameter. Each device reported similar values immediately after synchronization, and they report skewed values with time; values are more greatly skewed with further increases in time.

Fig. 12 shows that each of the devices synchronized at about 20 s. At the moment of synchronization, each device was synchronized with the gateway by executing the CoAP-based synchronization mechanism.

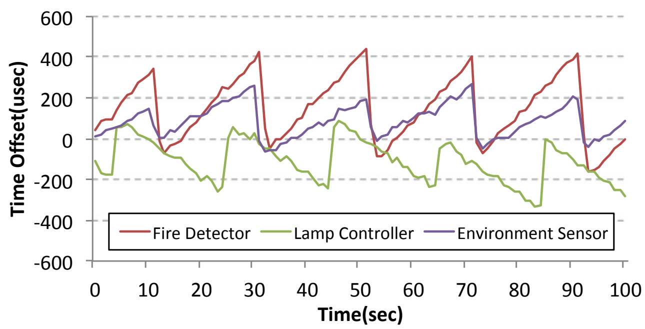
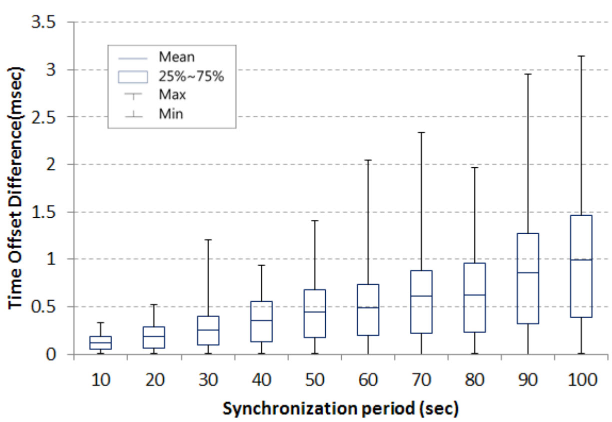
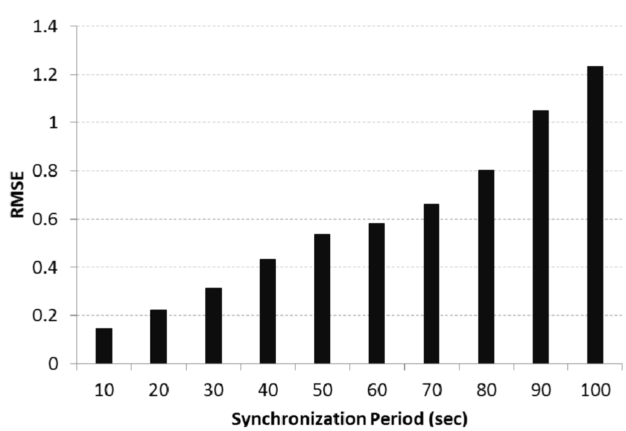


Fig. 12. Time offset sustaining a shape near zero obtained by the CoAP-based synchronization technique for each device.



(a)



(b)

Fig. 13. Analysis of the time synchronization accuracy according to the time synchronization period: (a) time offset difference vs. synchronization period, and (b) RMSE vs. synchronization period.

Fig. 13 shows the accuracy when using the CoAP synchronization technique. In the figure, the synchronization period changes from 10 s to 100 s. Fig. 13(a) shows the time offset differences that imply the maximum, minimum, median values, and 25–75 percentiles. Fig. 13(b) is the root mean square error (RMSE) per synchronization period. Both graphs show the dependency according to the synchronization period, i.e., as the synchronization period increases, the accuracy decreases. However, it is reasonable for the median and the maximum time offsets to be less than 1.5 ms and 3.5 ms, respectively, at a synchronization of about 100 s; for a 10 s synchronization period, an accuracy as high as about 0.12 ms can be obtained.

Considering that the exchange time for sensor data between the device and the home gateway is usually set to one seconds or longer intervals, the error of the proposed CoAP-based time synchronization technique is reasonable. In particular, the proposed scheme can be used in a variety of sensor networks because it is designed for a non-IP environment, so it can be applied to systems that cannot make use of existing time synchronization schemes, as is the case with NTP.

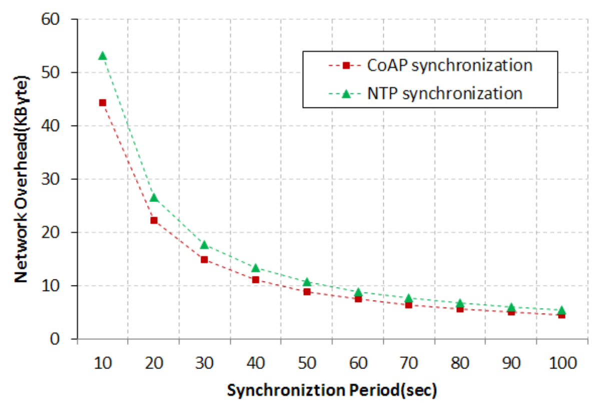


Fig. 14. Network overhead during 600 s obtained by synchronization period.

Frequent synchronizations logically increase the network overhead, so an appropriate tradeoff between system time accuracy and network overhead is critical to the design of a home automation system. To confirm this observation, Fig. 14 shows the network overhead for synchronization protocol messages and CoAP queries as the synchronization period changes from 10 s to 100 s. If only synchronization is considered, the synchronization with NTP may have the best performance, though if CoAP requests and responses coexist with NTP, the worst performance is observed. However, if the system exchanges CoAP messages to synchronize the time, the network overhead can be reduced by about 17% when compared to NTP when it includes CoAP queries. Since a piggyback method is utilized to synchronize the time on CoAP messages that are exchanged, the more the sensing data that is exchanged, the longer that the time remains accurate.

V. CONCLUSION

In general, household devices that support both a non-IP environment and an IP network are mixed within a home automation system. Accordingly, general protocols, such as TPSN, RBS, RTSP, and IEEE 1588, have been used to synchronize time in the home automation system. However, it remains difficult to effectively apply these protocols in wireless and non-IP environments that feature a limited communication interface having low processing performance and low power consumption because these conventional protocols are based on IP network access. Hence, a time synchronization technique was proposed here to support a non-IP environment that uses CoAP, a novel standard for M2M communications. The proposed scheme comparatively reduces network overhead because it only uses CoAP instead of the additional standards for time synchronization protocols. Another advantage is that it does not require an increase in performance, as experimental results indicate that the proposed scheme has a reasonable synchronization error when compared to NTP for existing distributed systems. Consequently, it is expected that the proposed CoAP-based time synchronization scheme can be extensively applied, not only in home automation systems but also in other applications, such as environmental monitoring, building and plant management, urban monitoring, disaster monitoring, etc.

REFERENCES

[1] F. Mattern and C. Floerkemeier, "From the Internet of Computers to the Internet of Things," *From active data management to event-based systems and more*, Springer Berlin Heidelberg, pp. 242-259, 2010.

[2] H. Gu, Y. Diao, W. Liu, and X. Zhang, "The design of smart home platform based on Cloud Computing," i*n Proc. International Conference on Electronic and Mechanical Engineering and Information Technology,* Harbin, CHINA, pp. 3919-3922, Aug. 2011.

[3] E. Spanò, L. Niccolini, S.D. Pascoli, and G. Iannaccone, "Last-Meter Smart Grid Embedded in an Internet-of-Things Platform,*" IEEE Trans. Smart Grid,* vol. 6, no. 1, pp. 468-476, Jan. 2015.

[4] O. Bergmann, K. T. Hillmann, and S. Gerdes, "A CoAP-Gateway for Smart Homes," *in Proc. International Conference on Computing, Networking and Communications*, Maui, USA, pp. 446 – 450, Jan. 2012.

[5] S. Y. Hwang and Y. J. Baek, "Reliable Time Synchronization Protocol in Sensor Networks," *Distributed Computing-IWDC*, vol. 3741, pp. 105-110, 2006.

[6] T. Kawai, N. Wakamiya, M. Murata, "Design and evaluation of a wireless sensor network architecture for urgent information transmission," *International Journal of Sensor Networks*, vol. 6. no.2, pp. 101-114, 2009.

[7] Y. S. Son, T. Pulkkinen, K. D. Moon, and C. Kim, "Home Energy Management System based on Power Line Communication," *IEEE Trans. Consumer Electron.,* vol. 56, no. 3, pp. 1380-1386, Aug. 2010.

[8] J. Han, C. S. Choi, and I. Lee, "More Efficient Home Energy Management System Based on ZigBee Communication and Infrared Remote Controls," *IEEE Trans. Consumer Electron.,* vol. 57, no. 1, pp. 85-89, Feb. 2011.

[9] H. C. Jo, S. Kim, S. K. Joo, "Smart heating and air conditioning scheduling method incorporating customer convenience for home energy management system," *IEEE Trans. Consumer Electron.,* vol. 59, no. 2, pp. 316-322, May. 2013.

[10] D. Mills, J. Martin, J. Burbank, and W. Kasch, "Network Time Protocol Ver. 4: Protocol and Algorithms Specification," IETF RFC 5905, 2010.

[11] S. Ganeriwal, R. Kumar, and M.B. Srivastava, "Timing-sync Protocol for Sensor Networks," *in Proc. International Conference on Embedded Networked Sensor System*, Los Angeles, USA, pp. 138-149, Nov. 2003.

[12] J. Elson, L. Girod, and D. Estrin, "Fine-grained Network Time Synchronization using Reference Broadcasts," *ACM SIGOPS Operating System Review*, 36(SI), pp. 147-163, 2002.

[13] M. Maroti, B. Kusy, G. Simon, and A. Ledeczi, "The Flooding Time Synchronization Protocol," *in Proc. International Conference on Embedded Networked Sensor System*, Baltimore, USA, pp. 39-49, Nov. 2004.

[14] K. Lee, J. C. Eidson, H. Weibel, and D. Mohl, "IEEE 1588-Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems," *in Proc. IEEE 1588 Conference*, Winterthur, Switzerland, Oct, 2005.

[15] T. Neagoe, V. Cristea, and L. Banica, "NTP versus PTP in Computer Networks Clock Synchronization," *in Proc. IEEE International Symposium on Industrial Electronics*, Montreal, Canada, vol. 1, pp. 317-362, Jul, 2006.

[16] Z. Shelby, K. Hartke, and C. Bormann, "The Constrained Application Protocol (CoAP)," IETF RFC 7252, 2014.

BIOGRAPHIES

Seung-Chul Son received the BS, the MS and PhD degrees in computer engineering from Chonnam National University, Gwangju, Korea, in 2002, 2007, and 2012, respectively. From 2005 to 2007, he was an R&D Engineer at the Grid Internet Broadcasting Research Center. Since 2008, he has been a senior member of engineering staff at Electronics and Telecommunications Research Institute (ETRI), Korea. His current research interests include calibration & synchronization techniques and mechanisms, and M2M, IoT technologies.

Nac-Woo Kim received the BS degree in control and instrumentation engineering from Chung-Ang University, Seoul, Korea, in 1997, and he received his MS degree and PhD degree in image engineering from Chung-Ang University, Seoul, Korea, in 2002 and 2006, respectively. He currently works as a senior member at the Electronics and Telecommunications Research Institute (ETRI), Korea. His research interests include video coding, image and video retrieval, and IPTV and 3DTV services

Byung-Tak Lee received the B.S. degree from Yonsei University, Seoul, Korea, in 1992, the M.S. and the Ph.D. degrees from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 1994 and 2000, respectively. From 2000 to 2004, he was a Principal R&D Engineer at LG Electronics, where he was engaged in the area of 1.6 Tbps long-haul Dense Wavelength Division Multiplexing (DWDM) system. Since 2005, he has been a Senior Research Engineer at Electronics and Telecommunications Research Institute (ETRI), Korea. His current research interests include fiber-to-the-home, passive optical networks, and Internet Protocol Television (IPTV) architecture.

Chae Ho Cho received the B.S. and M.S. degrees in computer engineering from Ajou University, Korea in 1997 and 1999, respectively. He was a Research Engineer at Unisec Company from 1999 to 2002. He was a Principal Researcher at Point-I Company from 2002 to 2007. He was also a Chief Researcher at M2Soft Company from 2007 to 2013. He is currently a Chief Researcher at M2soft Company, Korea. His research interests include mobile video system, mobile tele-presence system.

Jo Woon Chong received the BS, the MS and PhD degrees in electrical engineering from Korea Advanced Institute of Science and Technology, Daejeon, Korea, in 2002, 2004, and 2009, respectively. From 2010 to 2012, he was a postdoctoral fellow at the Massachusetts Institute of Technology (MIT). Since 2012, he has been a research assistant professor at the Worcester Polytechnic Institute, USA. His current research interests include cognitive radios systems, game theory, cross-layer design for communication systems, resource management, localization techniques, multiple access technologies for wireless local area networks and wireless sensor networks.