International Journal of Control and Automation
Vol. 11, No. 5 (2018), pp.163-172
http//dx.doi.org/10.14257/ijca.2018.11.5.14

A Study of IoT Home Network Management System Using SNMP

Jungyoung Han and Seunghyun Oh*

Computer Engineering Dept., Dongguk University Gyeongju Campus
{jyounghan, shoh}@dongguk.ac.kr

Abstract

In the age of the fourth industrial revolution, IOT-based home network systems are becoming popular. However, a home network incorporating a large number of various devices has not yet established a standardized device and protocol, and a methodology for managing a home network has not yet been established. In this study, we propose a management method by dividing various devices of home network into IP based and non-IP based devices. In IP based wireless sensor network, Publish/Subscribe communication system is implemented based on MQTT protocol, and ZigBee based communication system and management method is implemented and tested in non-IP based wireless sensor network. Traditionally, the Internet has been managed via SNMP by connecting all devices based on IP addresses. A home network containing many non-IP devices must be managed with a variety of items that are not managed by existing SNMP. This study proposes SNMP + sensor management method including comprehensive method and item considering this point. This allows user to visualize the sensor status to monitor at all times and respond to sensor failures, providing a reliable home network management solution.

Keywords: *Home Network, SNMP, Network Management, IoT, Zigbee*

1. Introduction

IoT has attracted a great deal of attention in the era of the Fourth Industrial Revolution, where large companies are investing in research and research by many researchers and companies, and with the adoption of the term "Internet of things" Market research institute Gartner predicts that 25 billion devices will be connected to wireless Internet-based networks by 2020 [1]. Recently, the Association for the Advancement of Mobile Phone Companies in Korea, it was found that "Healthcare" and "Smart Home" were the most energetic areas of the Internet in terms of Internet-related research [2].

In order to construct a smart home network, it is composed of many sensors based on object internet and data transmission is performed by various kinds of communication modules. However, the communication module is not standardized, and the research of sensor management is insufficient compared to the development of sensor technology. The transport protocols for the Internet include the Message Queuing Telemetry Transport (MQTT) [3], Extensible Message and Presence Protocol (XMPP), Representational State Transfer (REST) and Constrained Environments Application Protocol (CoAP) [4] It is a lightweight message protocol that enables asynchronous communication of limited resources, which are mainly used in home networks. CoAP is an application layer standard protocol based on 6LoWPAN [5] and is a protocol emerging in wireless sensor networks.

IoT 's various communication methods and sensor developments are evolving day by day, but the standard of various sensor management is inadequate, and the management of existing IP-based sensors is not always done before, in this paper,

Received (December 17, 2017), Review Result (March 10, 2018), Accepted (April 3, 2018)

* Corresponding Author

we propose a method to manage various non-IP sensors by using SNMP, regardless of the type of the communication module. And proposes a method that can be integrated and managed by the protocol. With the Proxy Agent for managing the SNMP, the data of the non-IP sensor and the IP-based sensor are transferred to the SNMP MIB (Management Information Base) The sensor data is sent frequently to the server's SNMP MASTER by registering sensor-specific data and the internal temperature and signal strength of the Arduino controller together. Experiments in this study have shown that integrated management of IP address-based device and ZigBee based sensor device using Wi-Fi has been completed effectively. For reference, this study is an extension of the concept for home network management proposed in [6].

In Section 2, we present a design for home network management based on SNMP. Section 4 presents the experimental results and finally Section 5 presents conclusions.

2. Related Studies

MQTT is a connection protocol for M2M (Machine-to-Machine) "Internet of Things". As a protocol developed by IBM in 1999, it has limited communication performance such as remote measurement and smartphone This is a protocol for transferring large-capacity messages designed for operation in a network environment. In 2013, only a message protocol for IoT is selected by OASIS (Organization for the Advancement of Structured Information Standards) But it has been verified to be an efficient product in terms of energy consumption of the battery [7].

A typical application of MQTT is collecting sensor information and activator control. The MQTT protocol enables sensor data to be constructed automatically with the gateway and it can be used as a subscription 1:N communication environment, the MQTT that supports 1:N communication over HTTP 1:1 communication minimizes the resource occupancy and satisfies the real-time property when the connected device is increased.

Sensors for home networks that use low power and limited memory are low in transmission cost and efficient MQTT is effective. However, since the MQTT protocol is a TCP/IP based communication protocol, only sensors that perform IP communication can be used. IBM developed the UDP-based MQTT-SN, a special version of MQTT aimed at sensor networks, that enables critical resource-constrained devices to use MQTT in untrusted networking environments [8].

The basic communication method of MQTT is shown in figure 1. The MQTT publishes and subscribes to topics of interest. Brokers such as *Mosquitto*, *HiveMQ*, *ver.X*, *ActiveMQ*, *Apollo*, *IBM MessageSight*, *JoramMQ*, *RabbitMQ*, and *Solace message Routers* are among the brokers. *Mosquitto* is an open source designed for use in limited devices with limited resources, especially in lightweight messaging situations. In this study, we will use *Mosquitto*, which fully supports MQTT, as an intermediary. All complex functionality is on the broker side and the client side is simple. *Mosquitto* uses port 1883 by default. Since each node of the MQTT based on IP knows only the address of the broker, it does not need to know the addresses of the other receiving nodes and it is not necessary to directly communicate with each other. Therefore, the broker can always transmit the value to the node requiring the value.

MQTT-SN is a protocol similar to MQTT, which is a topic-based IP environment developed by IBM, but performs non-IP based publish / subscribe functions. It transmits messages through Publish / Subscribe communication in wireless sensor environment like ZigBee instead of TCP/IP stack-based internet network. WSN

(Wireless Sensor Network) [9] has a bandwidth shorter than that of a wired link, and a link error rate is higher and lower than that of a wire. MQTT-SN is a WSN-optimized protocol with limited performance and battery. In the multi-hop range, messages are transmitted to the PtoP configuration capable of transmitting messages between two points based on the network address and UDP instead of connection-oriented within one hop range [10].

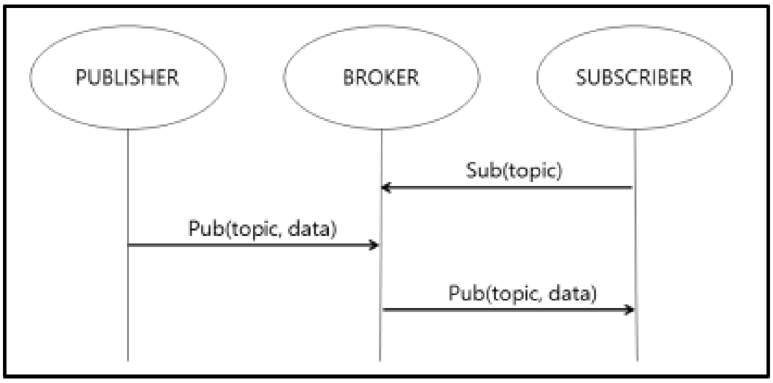


Figure 1. MQTT Topic based Publish/Subscribe Communication

Since each node of the MQTT based on IP knows only the address of the broker, it does not need to know the addresses of the other receiving nodes, and there is no need to directly communicate with each other, so the broker can always transmit the value to the node requiring the value. Communication must go through a broker, so if there are many nodes, broker overhead can occur. The QoS level [10] of MQTT is given, and the overhead can be reduced by receiving the value by changing the level to 0, 1 or 2 according to the importance level. The QoS level of MQTT is shown in the Table 1.

Table 1. QoS Level Classification of MQTT

MQTT QoS Level	Description
0	Up to once, data loss possible.
1	Delivery at least once, duplication possible.
2	Exactly once, no duplication.

Level 0 should be avoided if a value must be passed. Level 2 is assigned when delivering important messages whose value should not be lost. Since the requirement of Internet of objects is not enough for HTTP, MQTT is used in real-time information transmission through MQTT broker [11]. As shown in Figure 2, stable communication is possible, but sensor management is not mentioned.

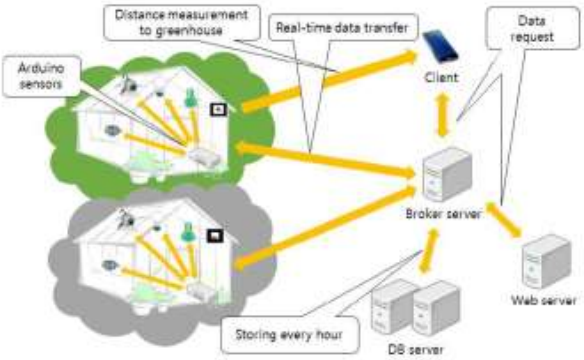


Figure 2. Schematic Diagram of the [11]

In a paper that introduces communication by the development of M2M technology, [12] introduces the superiority of standard protocols such as MQTT to the rapid development of Internet of things. Although many researches have demonstrated the superiority of MQTT as a communication method of Internet of things, research on the management aspects of sensors is still insufficient.

There are two types of MQTT-SN configuration: MQTT-SN Client and MQTT-SN Gateway. MQTT-SN client connects to MQTT server located in Internet through MQTT-SN Gateway. The MQTT-SN Gateway integrates or operates independently of the server, as well as data transfer between the MQTT and MQTT-SN protocols. According to the message conversion method of MQTT and MQTT-SN, it is classified as Transparent or Aggregating as shown in Figure 3.

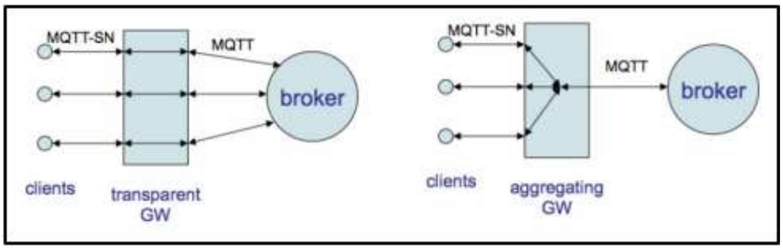


Figure 3. Transparent and Aggregation Gateway in the MQTT-SN

[13] proposed a solution that makes it possible to access the data of devices that cannot support SNMP. However, the efficiency of extensibility has been improved by designing the data storage and management functions as shown in Figure 4 Specific management measures that do not fit the characteristics are not presented.

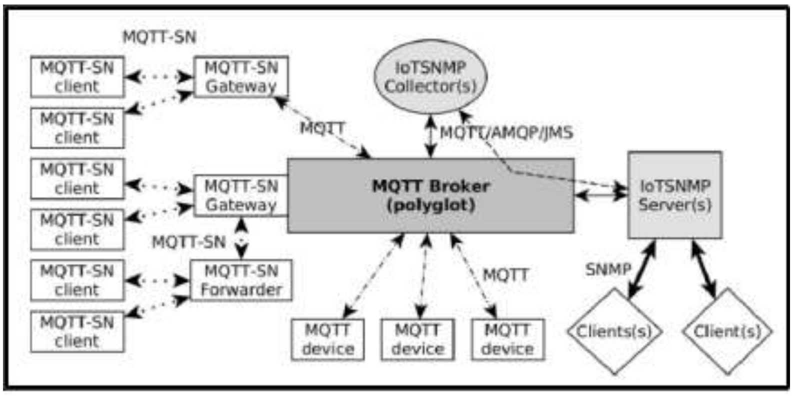


Figure 4. Overview of IoT SNMP Bridge in the [13]

There are libraries that support multiple languages with SNMP. *Net-SNMP* supporting C and Perl languages, *PySNMP* supporting Python language, and *SharpSNMP*[14]supporting C # language. *PySNMP*[15]consists of a small number of stand-alone components with well-defined interfaces as shown in Figure 5.

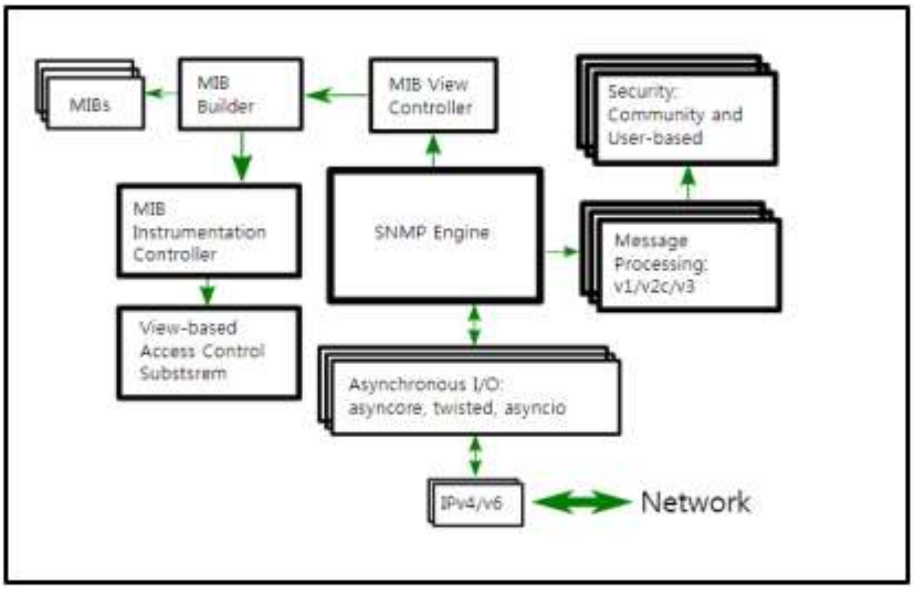

MIBs
MIB Builder
MIB View Controller
Security: Community and User-based
MIB Instrumentation Controller
SNMP Engine
Message Processing: v1/v2c/v3
View-based Access Control Substsrem
Asynchronous I/O: asyncore, twisted, asyncio
IPv4/v6
Network

Figure 5. PySNMP Function Structure

The *SnmpEngine* class is a core object of SNMP. There is a unique identifier that can be assigned to management, and this identifier is used for SNMP protocol operations. In this paper, we will use *Pysnmp*.

3. A SNMP-based Sensor Management System Design

As shown in Table 2, the transmission module and the transmission method should be applied differently according to the sensor application field. The transmission method is classified into a sensor transmitting periodically and a sensor transmitting when an event occurs, and the communication module is classified also considering power consumption.

Table 2. Communication Module and Transmission Method for Each Sensor

Site	Sensor	Period	Event	Publish	Subscribe	Comm.
Room1	Temp.	Yes		Yes		ZigBee
	Humidity	Yes		Yes		ZigBee
	Illumination	Yes		Yes		WI-FI
Room2	Gas		Yes	Yes		Z-Wave
	Smoke		Yes	Yes		ZigBee
	Sprinkler		Yes		Yes	WI-FI
Room3	Door-Lock		Yes		Yes	NFC or RFID
	Infrared		Yes		Yes	ZigBee

The in this section, the system is designed and described as follows to enable the management of non-SNMP-based and non-IP-based sensors for TCP/IP-based sensors. The MQTT, which is subject to the legend of the IoT communication protocol, uses a MQTT for the home network with a limited communication bandwidth and suitable for the sensor, and for the TCP/IP-based sensor communication. We used *Mosquitto 1.4.9* for the configuration [16]. As an example of the non-IP address and communication module, ZigBee communication module was used. ZigBee is a typical example of low power consumption and communication protocol, are used in various sensor devices.

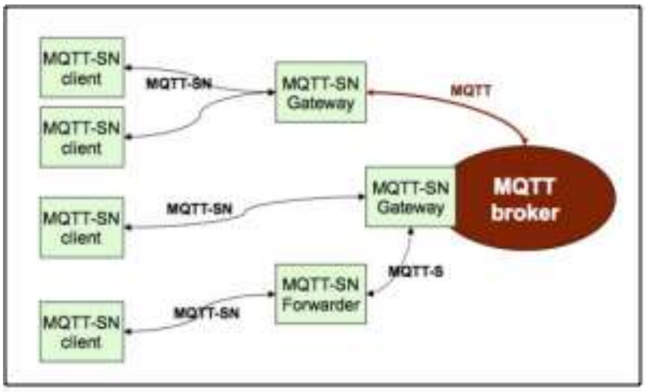


Figure 6. An Example Structure of MQTT-SN Connection Over Non-IP Sensor Devices; MQTT-SN Gateway Collects Sensor Information Instead of MQTT Broker

The structure presented for the integration of IP address-based devices and non-IP addressed devices in SNMP is SNMP Proxy Agent. In this research, we modified SNMP Proxy Agent related part so as to manage all sensor devices used in smart home network. The existing Proxy Agent is named SNMP+Sensor after registering necessary parts for sensor device management anew and registers sensors not in SNMP management items.

Information on the value of the transmitted sensor will be stored in the MIB of the Agent. Save the internal temperature and signal strength of the added controller that is not supported together with the basic information of the sensor supported by SNMP together with the sensor value. The information stored in the MIB will be forwarded if there is a request from the MASTER of the server and will be sent to the TRAP when the event occurs.

To integrate IP-based sensor information with non-IP-based sensor information, Proxy Agent is used. The SNMP+Sensor of the proxy agent registers the sensor which is not in the management item of SNMP. The received sensor value and information are stored in the Agent's MIB. It is not supported with basic information of sensors supported by SNMP, so the added controller internal temperature and signal strength are stored together with the sensor value. The information stored in the MIB is forwarded to the server's MASTER if requested or to the TRAP when the event occurs. You must specify the name of the MIB object and how the object is to be managed, and modify *snmpd.conf* to allow extended SNMP to work. The UDP port is set to use the default 161. Add the IP of the administrator who wants to receive the TRAP and TRAP the changes to the administrator.

The SNMP Agent parses the data received from the MQTT broker and sensor values based on the ZigBee to enable SNMP trapping to the main server. The proxy agent information is SNMP TRAP to the main server. The transmitted value is parsed according to the OID value. Basic management items of SNMP and sensor management items added to SNMP+Sensor are all stored in the database for the purpose of integrated management.

4. Experiments and Results

This chapter describes the overall system implementation for the home network system to be implemented and the final monitoring of the sensors through experiments. The proposed system configuration is shown in Figure 7.

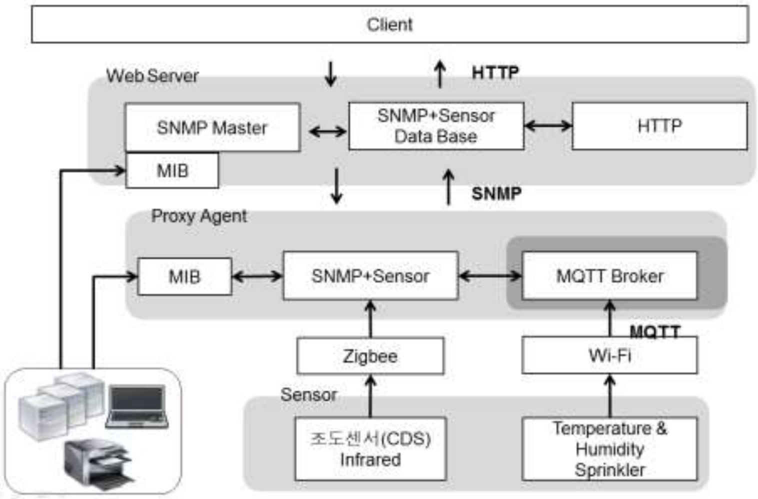


Figure 7. System Configuration

SNMP provides basic management of IP-based devices. Systems designed to manage SNMP and non-IP sensors are divided into Sensor group, Proxy Agent group, and Web Server group.

A. Sensor Group

- *IP-based Sensor: MQTT protocol for Wi-Fi communication*
- *Non-IP based sensor: ZigBee communication with minimum power consumption*

B. Proxy Agent Group

- *SNMP + Sensor: Input sensor information by type and send it to web server with SNMP trap message*

C. Web Server Group

- *Parse OID information and store it in database*
- *Build website after PHP work for monitoring stored data*

D. Client

- *Real time monitoring with HTTP enables to deal with sensor failure*

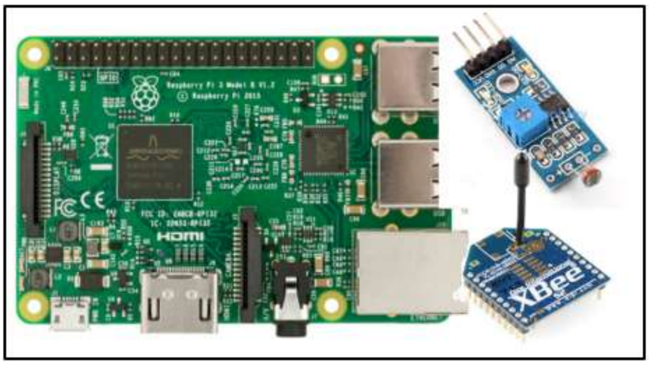


Figure 8. An Example of Non-IP ZigBee Sensor of *XBee*

In our study, SNMP + Sensor, which is a proposed structure for integrated management of IP-based devices and non-IP-based devices, was implemented as a database through investigation of management items as follows. We will use open source MySQL, one of the relational database management systems. MySQL provides Python and PHP APIs for web pages. Table 3 shows the table structure for storing sensor data values.

Table 3. SNMP MIB Data Table

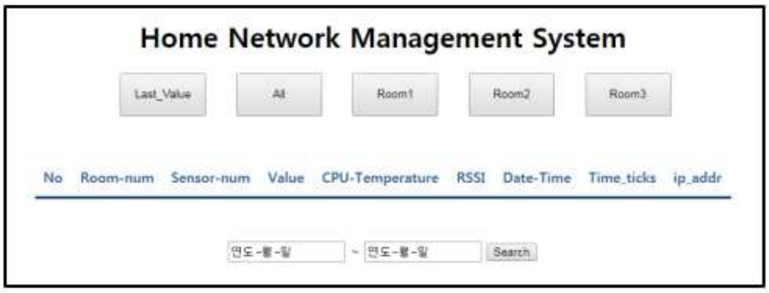
Field	Type	Null	Key	Default
Type	varchar(8)	YES	PK	null
Sensor_val	float(8)	YES		null
Sensor_val2	float(8)	YES		null
Strength	int(10)	YES		null
Tmp_mcu	float(8)	YES		null
Time_Ticks	varchar(64)	YES		null
Ip_add	varchar(64)	YES		null
Creation_time	datetime default current_timestamp			

The source code below is a code that PUBLISHs the temperature, humidity, controller internal temperature, and signal strength (RSSI) of the sensor node every 5 seconds to the MQTT broker as shown in Figure 9.

```
String payload = "{";
  payload += "\"sensor_val\":"; payload += temperature; payload += ",";
  payload += "\"sensor_val2\":"; payload += humidity; payload += ",";
  payload += "\"tmp_mcu\":"; payload += String(GetTemp()); payload += ",";
  payload += "\"strength\":"; payload += String(getRSSI());
  payload += "}";
  char attributes[100];
  payload.toCharArray( attributes, 100 );
    mqttclient.publish("test",attributes);
  }
  delay(5000);
}
```

Figure 9. Code for Sensor Value Publish to MQTT Broker

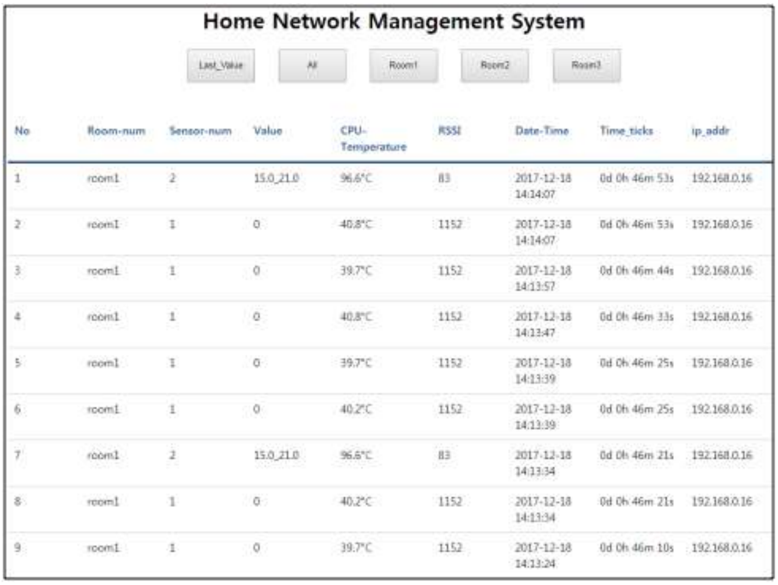
The user can monitor the status of the sensor by enabling real-time monitoring. Sensor failure, connection loss, sensor node overload status, *etc.*, can be detected, and if there is an error, it can be coped quickly. The monitoring screen consists of an initial screen, a recent value, and the last 100 values to monitor the required status like below three figures.



Home Network Management System

No	Room-num	Sensor-num	Value	CPU-Temperature	RSSI	Date-Time	Time_ticks	ip_addr

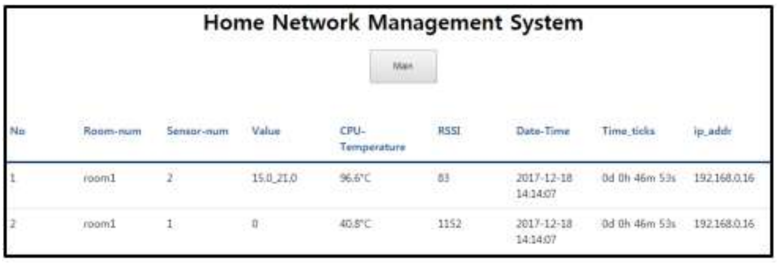
Figure 10. Initial Web Service Screen



Home Network Management System

No	Room-num	Sensor-num	Value	CPU-Temperature	RSSI	Date-Time	Time_ticks	ip_addr
1	room1	2	15.0_21.0	96.6°C	83	2017-12-18 14:14:07	0d 0h 46m 53s	192.168.0.16
2	room1	1	0	40.8°C	1152	2017-12-18 14:14:07	0d 0h 46m 53s	192.168.0.16
3	room1	1	0	39.7°C	1152	2017-12-18 14:13:57	0d 0h 46m 44s	192.168.0.16
4	room1	1	0	40.8°C	1152	2017-12-18 14:13:47	0d 0h 46m 33s	192.168.0.16
5	room1	1	0	39.7°C	1152	2017-12-18 14:13:39	0d 0h 46m 25s	192.168.0.16
6	room1	1	0	40.2°C	1152	2017-12-18 14:13:39	0d 0h 46m 25s	192.168.0.16
7	room1	2	15.0_21.0	96.6°C	83	2017-12-18 14:13:34	0d 0h 46m 21s	192.168.0.16
8	room1	1	0	40.2°C	1152	2017-12-18 14:13:34	0d 0h 46m 21s	192.168.0.16
9	room1	1	0	39.7°C	1152	2017-12-18 14:13:24	0d 0h 46m 10s	192.168.0.16

Figure 11. An Example of User Interface Screen with Last 100 Values



Home Network Management System

No	Room-num	Sensor-num	Value	CPU-Temperature	RSSI	Date-Time	Time_ticks	ip_addr
1	room1	2	15.0_21.0	96.6°C	83	2017-12-18 14:14:07	0d 0h 46m 53s	192.168.0.16
2	room1	1	0	40.8°C	1152	2017-12-18 14:14:07	0d 0h 46m 53s	192.168.0.16

Figure 12. The Screen shows Last Captured Value

5. Conclusions

The first SNMP approval for IP-based sensor management was in 1990. The development of non-IP communication in a network environment that develops differently is also surprising. IP / Non-IP communication is used for the communication of IoT devices, but the standardization of the management of devices is still insufficient. Through this study, we applied SNMP to the management of IoT devices and added items that cannot be managed by SNMP, which greatly contributed to the management of IoT equipment.

In this paper, SNMP, which is the management protocol of IP based sensor, was used for monitoring for managing faults of various sensors constituting the home network. We also proposed a scheme to enable management of sensors and non-IP based sensors that SNMP cannot manage. We confirmed that heterogeneous sensor devices of smart home network can be interestedly managed via SNMP+Sensor proposed in experimental environment implemented together with various sensor devices. The collected information can be used to predict signs of failure of the sensor device through analysis of big data. Also, by knowing the current state of the device, it is possible to effectively manage a small sensor, especially when a large number of sensor devices are installed in the home environment.

Acknowledgments

This paper is a revised and extended version of a paper “A Research on Sensor Management Scheme in the Home Network” presented at 10th AST 2018, Ho Chi Minh, Vietnam, Feb 1-3, 2018.

References

[1] Gartner, Inc. Gartner Says 6.4 Billion Connected “Things” Will Be in Use in 2016, Up 30 Percent From 2015 [Internet] Available: http://www.gartner.com/newsroom/id/3165317.

[2] Korea Mobile Enterprise Promotion Association, www.kmea.or.kr, **(2014)**.

[3] MQTT, http://mqtt.org

[4] Z. Shelby, K. Hartke and C. Bormann, “The constrained application protocol (CoAP)”, RFC7252, **(2014)**.

[5] Z. Shelby and C. Bormann, “6LoWPAN: The wireless embedded Internet (Vol. 43)”, John Wiley & Sons. **(2011)**.

[6] J. Han and S. Oh, “A Research on Sensor Management Scheme in the Home Network”, Proceedings of the 10th International Conference on Advanced Science and Technology, Ho Chi Minh, Vietnam, **(2018)** February 1-3.

[7] S.-h. Kim, D.-h. Kim, H.-s. Oh, H.-s. Jeon and H.-j. Park, “The Data Collection Solution Based on MQTT for Stable IoT Platforms”, Journal of the Korea Institute of Information and Communication Engineering, vol. 20, no. 4, **(2016)**, pp. 728-738.

[8] A. Stanford-Clark and H. L. Truong, “Mqtt for sensor networks (mqtt-sn) protocol specification”, International business machines (IBM) Corporation version, 1, **(2013)**.

[9] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey”, Computer networks, vol. 38, no. 4, **(2002)**, pp. 393-422.

[10] S.-J. Kim, K.-W. Cho, M.-E. Lee and C.-H. Oh, “A Study on Adaptive QoS Control System based on MQTT for Reducing Network Traffic”, International Conference On Future Information & Communication Engineering, vol. 9, no. 1, **(2017)**, pp. 223-226.

[11] D.-E. Kim, S.-W. Kim and S.-K. Kwon, “Real-Time Transmission System for Greenhouse Information Using MQTT and RTSP”, Journal of Korea Multimedia Society, vol. 18, no. 8, **(2015)**, pp. 935-942.

[12] S.-H. Shim and H.-B. Kim, “Internet of Things and MQTT technology”, Review of KIISC, vol. 24, no. 6, **(2014)**, pp. 37-47.

[13] M. Savić, “Bridging the SNMP gap: Simple network monitoring the internet of things”, Facta universitatis-series: Electronics and Energetics, vol. 29, no. 3, **(2016)**, pp. 475-487.

[14] sharpSNMP, web site: https://github.com/lextm/sharpsnmplib.

[15] pySNMP, web site: pysnmp.sourceforge.net

[16] An Open Source MQTT v3.1/v3.1.1 Broker, https://mosquitto.org.

Authors

Jungyoung Han, she received M.S. degree in computer science from Dongguk Univeristy, Gyeongju, Korea in 2018. Her search interests are IoT, Sensor Networks, Home Networings.

Seunghyun Oh, he received the B.S. degree in computer science from Dongguk University, Seoul, Korea, in 1988, and the M.S. and Ph.D. degrees in computer engineering from Dongguk University, Seoul, Korea, in 1998 and 2001, respectively.

He has been a Professor with the Department of Computer Engineering, Dongguk University Gyeongju Campus since 2002. His current research interests include wireless communications and sensor networking system.