International Journal of Energy, Information and Communications
Vol.11, No.1 (2020), pp.15-22
http://dx.doi.org/10.21742/ijeic.2020.11.1.03

Comparison of Power Consumption According to the Communication Protocol Mode of Fine-Dust IoT Device Based on OneM2M Standard Protocol

Seong-Se Cho[1], Seung-Hun Kim[2], and Won-Hyuck Choi[3]

[1,2]Dept. of Aeronautical Systems Engineering Hanseo University, 236-49, Gomseom-ro, Nam-myeon, Taean-gun, 32158 Chungcheongnam-do, Republic of Korea
[3]Department of Avionics Engineering Hanseo University, 236-49, Gomseom-ro, Nam-myeon, Taean-gun, 32158 Chungcheongnam-do, Republic of Korea
[1]choiwh@hanseo.ac.kr, [2]chosse63a@gmail.com, [3]kimsh014@gmail.com

Abstract

As all things are connected to the Internet, the Internet of Things has become possible through communication without user intervention, and the Internet of Things has recently focused on various areas such as smart homes, smart cars, smart factories, and industrial sites. In the HTTP communication protocol environment, we established a system to communicate between client servers using the communication protocols MQTT and CoAP communication protocols together with the fine dust sensor and confirmed the amount of power generated according to the message transmission mode used for each communication.

Keywords: *OneM2M, MQTT, CoAP, Standard protocol, Fine dust, IoT*

1. Introduction

With all things connected to the Internet, interaction is possible through communication between objects without user intervention, and the Internet of Things has recently been paying attention to various areas such as smart homes, smart cars, smart factories and other industrial sites such as manufacturing, agriculture, and construction [1].

According to market research institutes and experts, the size of the related IoT market is expected to grow in 2020, with 50 billion Internet of Things devices from around the world connected. In order to take the initiative in the IoT ecosystem, labor is being done to pre-empted patents and standards through fierce technology development. Things use sensors to identify various behaviors or changes in the environment around things that we do in our daily lives or social activities and send them via the Internet as digital data. This means that digital copies of our real world are made in the cyber world, or the cloud. As the number of objects connected to the Internet increases, more precise digital copies of the real world are made [2] [3].

The IoT industry can be divided into platforms, services, networks, security and devices [4] [5].

First of all, the platform field is where global giants such as Google, Microsoft, and SKT are striving to become the leaders in the IoT ecosystem, but there are no operators who have

Article history:
Received (October 5, 2019), Review Result (December 24, 2019), Accepted (January 12, 2020)

Print ISSN: 2293-9655, eISSN: 2652-1989 IJEIC

dominated the IoT platform yet, and small and medium-sized companies are hesitant to develop and enter the market due to the sluggishness of open platforms. The service sector is changing from a previous environment using a handset to an open ecosystem where smartphones can be developed and utilized by anyone [6].

Devices that use IoT have limited resources and IoT devices will have small resources and limited communication environments with demands for long hours of use. Light communication protocols are needed for communication of such devices and protocols such as MQTT and CoAP, which can be used in IoT, are receiving attention. We want to compare the efficiency of the two protocols used in IoT.

In this paper, we are going to construct a communication environment between devices using MQTT and CoAP, which are communication protocols designated for Internet of Things on the oneM2M platform using Arduino and fine dust sensors among open-source hardware, and analyze and compare the amount of power consumed by each communication protocol, focusing on HTTP, which is an Internet communication protocol.

2. Implementation

In this chapter, the system is deployed to analyze the amount of power consumed by the protocols described earlier. The client developed a communication program using CoAP, MQTT, and HTTP protocols on the ATmega328 board using the Arduino IDE development tool. The client uses the external power to verify the power used every 30 seconds and uses each communication protocol to transmit the measured amount of power to the server. Measure the voltage and current to check the external power supply. [Figure 1] shows a client hardware configuration diagram [7][8].

In the client, the MQTT protocol was developed using <PubSubClient.h> in the PubSub Client library. When communicating using the MQTT protocol, proceed through the MQTT broker. The client sends the data of fine dust measured in /dust Topics to the broker, and the server program proceeds with the subscription to that topic. The broker sends data from the client to the server that has subscribed to /dust topics [9].

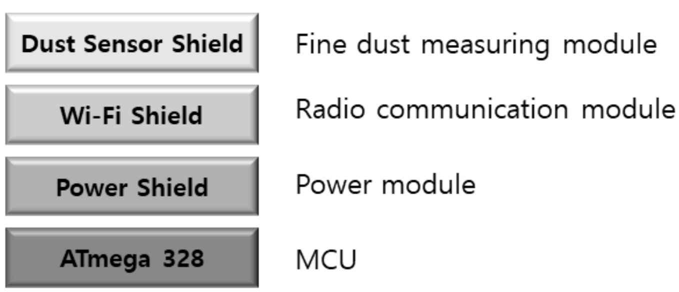

Figure 1. Client hardware configuration diagram

MQTT clients connect to the gateway using the MQTT protocol to access the Public/subscribe service of the MQTT broker server located on the wireless network and present on the external Internet network. Currently, the gateway that connects with the MQTT client functions by connecting to the broker server through the external Internet network. In addition, the gateway consists of an integrated or independent form with the broker server, depending on the application, and when it operates in an independent form, the gateway communicates with the broker server using the MQTT protocol [10].

The sensor node is an and device that forms an MQTT client and Wi-Fi mesh network, which is connected to a gateway to transmit sensor data or perform a Public/subscribe function to control the actuator. Depending on the functions performed in this system, the Sensor Node consists of three types of nodes that perform all the functions of the Public communication function, the Subscribe communication function, and the Public and Subscribe. The MQTT protocol supports multiple MQTT gateways in a single wireless sensor network. Thus, the gateway consists of routers in the Wi-Fi mesh network, serves as a connection between the external Internet network and the Wi-Fi-based wireless network, and is tasked with converting and delivering messages to and from the format of the MQTT protocol [13][14]. [Figure 2] shows the flow of data when using the MQTT protocol.

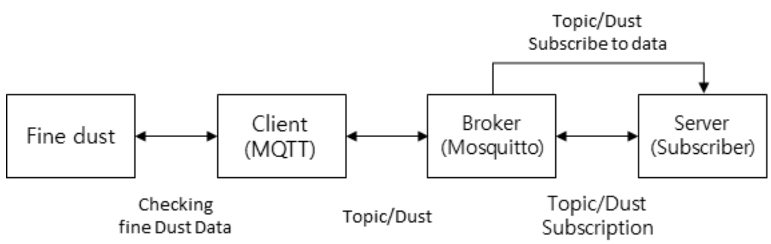

Figure 2. Data flow with MQTT protocol

The CoAP protocol client program was developed using the CoAP-Simple-library library's <coap.h>. The CoAP protocol is a protocol that operates on a UDP basis, and CoAP can establish reliability and non-reliability transmission among data types [11].

A CoAP server, unlike an HTTP server, is a server that is substantially connected to objects in a limited resource environment. Each object will communicate based on the IEEE 802.15.4 standard, just like a network architecture. The communication radius of the IEEE 802.15.4 standard, which is based on low power and low cost, will be around 10 m and space within 10 m is referred to as the Personal Operating Space (POS). When objects with support for wireless networks enter the POS, they will be part of the user's Wireless Personal Area Network and will communicate data [12].

First, it's a way of searching for things. Each CoAP server communicates with objects in a wireless network environment, and each server has a resource directory. The Resource Directory is a list of things that contains information about them and explores each object in a resource directory. The path to the resource directory is specified as /. well-known/core and you can use queries to navigate through the resources you want. There is a lookup feature in the resource directory that explores objects or groups. The lookup function calls using the GET method of the REST service and allows you to explore the type of resource you want. It may be called by designating objects or groups according to type. It can also be called for a few specific objects or groups, not just a few objects or groups.

Perform the Cooper installation in the main browser as shown in [Figure. 3].

The server program generates classes of each communication protocol as it works, operates each server, and waits for the client's response. MQTT makes a subscription to /dust Topics to a pre-powered broker and receives and processes events from the client when the data on that topic occurs.

The server program is always waiting for the access of clients using each communication protocol, and if data occurs from the client, the data is post-processed and then entered into the database.

Figure 3. Browser settings

To verify the power usage for each communication protocol, three clients were operated simultaneously, and external power was authorized. Each client program continues to check external power, check communication protocols to the server, and send external power data.

[Figure 4] serves as a server and storage of data provided by MQTT, CoAP, and HTTP protocols based on the dotnet framework.

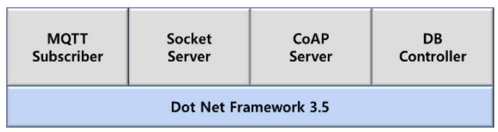

Figure 4. Server system configuration diagram

3. Results and analysis

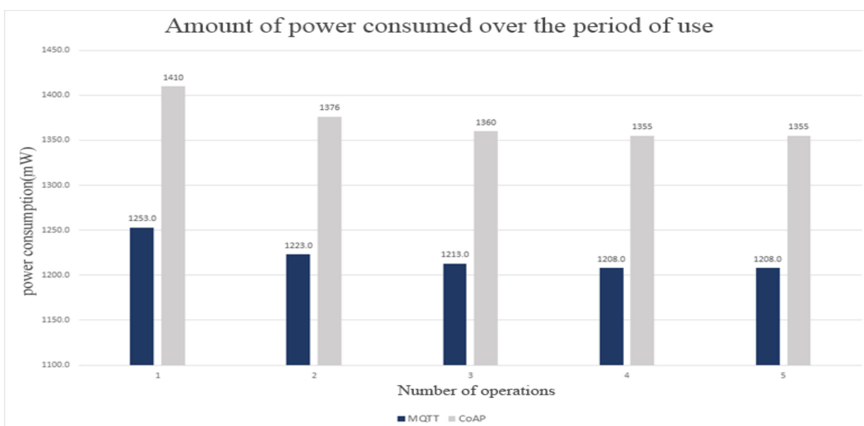

Figure 5. Amount of power consumed over the period

The message issuance cycle was increased from 1 second to 5-second intervals to compare the power consumption of the MQTT and CoAP protocols, with a message size of 2 bytes. As shown in [Figure 5], the consumption power generated per second was measured at approximately 1260mW for the MQTT and the largest consumption power at 1405mW for the CoAP. The 3-second cycle shows a similar amount of power consumption as the 5-second cycle.

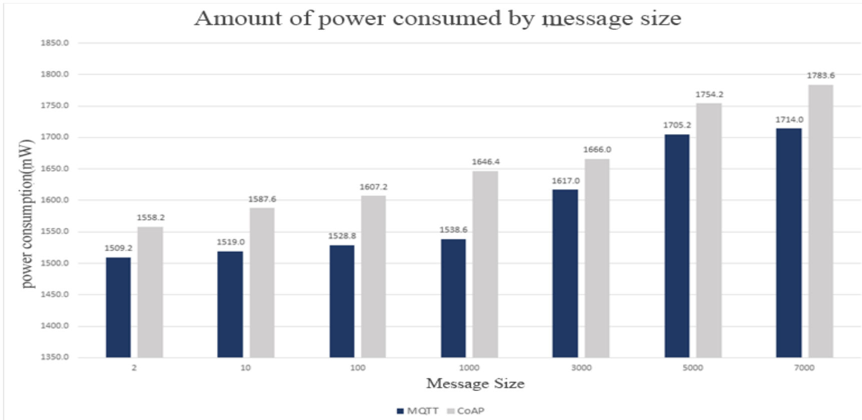

Figure 6. Amount of power consumed by message size

As shown in [Figure 6], if the message size is less than 100 bytes, the average power consumption varies by about 65mW on average. CoAP increases to about 107.8 mW on 1000 bytes, increases to 1705.2mW and 1754.2mW about 49mW on 5000 bytes, and 1714.0mW and 1783.6mW on 7000 bytes.

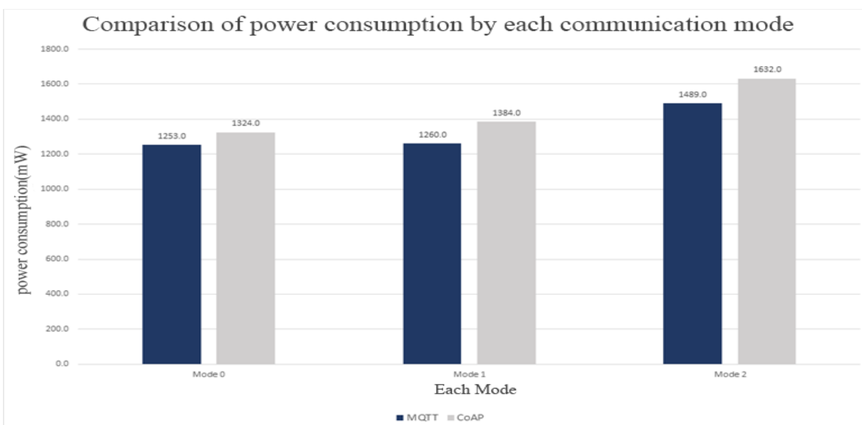

Figure 7. Comparison of power consumption by each communication mode

The measurement results are shown in [Figure 7]. In Mode1, the difference in power consumption between MQTT and CoAP is 71mW. In Mode2, the difference in power consumption between MQTT and CoAP is 124mW. In Mode3, the difference in power consumption between MQTT and CoAP is 143mW [13].

4. Conclusion

The message issuance cycle was increased from 1-second to 5-second intervals to compare the power consumption of the MQTT and CoAP protocols, with a message size of 2 bytes. The amount of power consumed in 1 second was measured at approximately 1260mW for MQTT and 1405mW for CoAP, showing the largest amount of power consumed. The 3-second cycle shows a similar amount of power consumption as the 5-second cycle.

If the message size is less than 100 bytes, the average power consumption varies by about 65mW on average, and CoAP increases to about 107.8 mW on 1000 bytes, 1705.2mW and 1754.2mW on 5000 bytes, and 1714.0mW and 1783.6mW on 7000 bytes.

In MQTT, it was found that 71mW for non-response at QoS 0 and CoAP and 124mW for a general response from QoS 1 and CoAP were consumed slightly at MQTT. In MQTT, we could see an increase in power consumption at 143mW when piggy backed at QoS 2 and CoAP.

In environments such as sensing data that simply transmits measured data, such as ambient atmospheric environments, the MQTT message type was shown to provide better results than previously used HTTP communication protocols in environments where the use of MQTT message types is appropriate and cannot reliably power, and where consumption should be prioritized if long hours of availability should be guaranteed with low power.

Acknowledgements

This study was supported by a 2019 research grant from Hanseo University and was supported by the Korea Institute for Advancement of Technology (KIAT) grant funded by the Korea government (MOTIE: Ministry of Trade, Industry & Energy) (No. N0002431)

References

[1] M. Prihodko, “Energy consumption in location sharing protocols for Android applications,” M.S. thesis, Linköopings University, **(2012)**

[2] J. S. Myeong, “IoT service platform standardization status,” TTA Journal, **(2016)**

[3] S. Y. Kim and K.Y. Kim, “Standardization of IoT service platform,” vol.6, no.32, **(2014)**

[4] oneM2M, oneM2M Service Layer Protocol Core Specification, **(2014)**

[5] Milan Tucic, “Networking layer for unifying distributed smart home entities,” 22nd telecommunications forum telFOR, November 25-27, Belgrade, Serbia, **(2014)**

[6] M. Dong, T. Lan, and L. Zhong, “Rethink energy accounting with cooperative game theory,” MobiCom 14: Proceedings of the 20th annual international conference on Mobile computing and networking, September pp.531-542, New York, United States, **(2014)**

[7] H. S. Kim, S. Kumar, and D. E. Culler, “Thread/Open thread: A compromise in low-power wireless multihop network architecture for the internet of things,” IEEE Communications Magazine, vol.57, no.7, pp.55-61, **(2019)**

[8] X. Liu and F. Qian, “Rethink energy accounting with cooperative game theory,” MobiCom 14: Proceedings of the 20th annual international conference on Mobile computing and networking, September, pp.531-542, Maui Hawaii USA, **(2014)**

[9] Message Queuing Telemetry Transport, **(2014)**

[10] A. Pathak, Y. C. Hu, M. Zhang, P. Bahl, and Y. M. Wang, “Fine-grained power modeling for smartphones using system call tracing,” EuroSys 11: Proceedings of the sixth conference on Computer systems, April, pp.153-168, New York, United States, **(2011)**

[11] A. Pathak, Y. C. Hu, and Zhang. “Where is the energy spent inside my app?: fine grained energy accounting on smartphones with prof,” Proceedings of the 7th European Conference on Computer Systems, April, pp.29-42; Bern, Switzerland, **(2012)**

[12] S. Kim, D. Kim, H. Oh, H. Jeon, and H. Park, “The data collection solution based on MQTT for stable IoT platforms,” J. Korea Inst. Inf. Commun, Eng, vol.4, no.20, **(2016)**

[13] C. Berrhouma, A. Elmangoush, A. A. Hazmi, R. Steinke, and T. Magedanz, “Performance evaluation of an m2m platform in different deployment setups,” Proceedings of 2016 IEEE First International Conference on Internetof-Things Design and Implementation, May 19; Berlin, Germany, **(2016)**

Authors

Seongse Cho
Received his bachelor's degree in computer science from the Korea National Open University in 2003. In 2017, he received the Master degree in avionics engineering from Hanseo University, Seosan, Korea. His research interests include military fighter jet avionics system, UAV collision avoidance, robotics, and intelligent transportation systems.

Seung-Hun Kim
He is a master's degree in Aeronautical System Engineering, Hanseo University. He graduated from the Department of Aeronautical and Mechanical Engineering, Hanseo University. He is studying Kalman filters, embedded systems, and smart network communications.

Won-Hyuck Choi
He received the Ph.D. degree in avionics from Korea Aerospace University, Korea. He was a Professor of smart network at Doowon Technical University College, Korea. urrently, he is Professor of avionics at Hanseo University, Korea, where he has been since 2014. His present research interests include embedded system, home network.

This page is empty by intention.