Comparison with HTTP and MQTT on Required Network Resources for IoT

Tetsuya Yokotani
Department of Electronics, Information and Communication Engineering, College of Engineering
Kanazawa Institute of Technology
Nonoichi, Ishikawa, Japan
yokotani@neptune.kanazawa-it.ac.jp

Yuya Sasaki
Department of Electronics, Information and Communication Engineering, College of Engineering
Kanazawa Institute of Technology
Nonoichi, Ishikawa, Japan
b1321780@planet.kanazawa-it.ac.jp

***Abstract*— HTTP has been widely applied for data transfer. However, in networks for IoT, this protocol causes a large overhead. To solve this problem, named based transfer protocols have been discussed. This paper compares the performance of HTTP with that of MQTT, a type of named based transfer protocol. Additionally, the paper proposes enhancements to MQTT for better performance.**

Keywords—IoT, HTTP, MQTT, NDN, ICN, Performance Evaluation, Protocol Overhead

I. Introduction

Internet of Things (IoT) is being widely discussed. It is a topic of worldwide interest. In IoT, a large number of tiny data blocks from devices, such as various sensors, are transferred across networks. Although Internet Protocol (IP) has been adopted for most types of communication, it will have some problems when it is applied to IoT.

Currently, Internet access requires application protocols over TCP/IP or UDP/IP. One of the application protocols is Hyper Text Transfer Protocol (HTTP), which has been standardized in IETF, e.g., [1] (initial version) and [2] (the latest version), and has been applied for general communication over Internet. However, when HTTP is applied to communication in IoT, in which a huge number of tiny data blocks are transferred, protocol overhead and resulting performance degradation are a serious problem. Moreover, IP addressing depends on physical location, which causes the problem of complexity of network control. To solve these problems, name- based architectures, such as Named Data Networking (NDN), Content Centric Networking (CCN), and Information Centric Networking (ICN) have been discussed; see e.g., [3] – [9]. Some of the examples focus on adopting these architectures to IoT; see e.g., [10] — [12].

In these architectures, MQ Telemetry Transport (MQTT) is one of the protocols, as described in [13]. Standard committees such as oneM2M and ETSI have paid considerable attention to MQTT and have also conducted relevant discussions. MQTT reduces protocol overheads and provides high efficiency communication for IoT. It also invokes "Name based routing," and mitigates IP address based routing for IoT traffic flows.

This paper discusses the possibility of considering MQTT as a candidate for the communication protocols on the IoT platform. It compares the performance of MQTT with that of HTTP. Moreover, it proposes new mechanisms to enhance the current MQTT specifications.

II. Architecture of IoT and Communication Protocols

As described in the previous section, IoT is being actively discussed worldwide. National base activities, such as "Industry 4.0" in Germany, trigger off such heated discussions.

IoT includes various services as a social infrastructure. Typical examples of IoT applications are shown in Figure 1.

Fig. 1. Typical applications in IoT

To provide these services, the IoT architecture is of two types, "Vertical" and "Horizon," as shown in Figure 2. This point is mentioned in many papers and articles, e.g. [14].

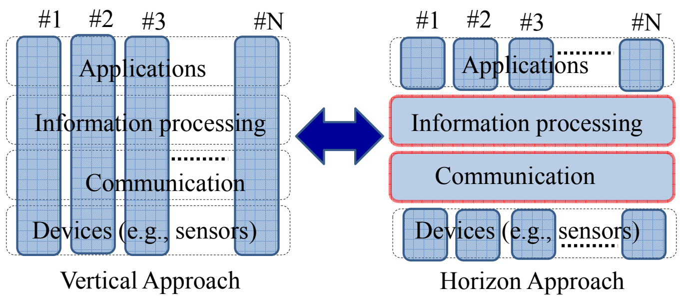

Fig. 2. Architecture on IoT

In the Vertical approach, Information processing and Communication functions can be optimized for every application. However, these functions have to be deployed individually. In comparison, in the Horizon approach, Information processing and Communication functions are shared across all the applications. Since most of the applications in IoT are provided across a wide area, they expect Horizon approach.

Communication protocols in the Horizon approach have to be based on IP and its related protocols for coexistence with the legacy communication services. Therefore, the protocol stack for IoT communication is shown in Figure 3.

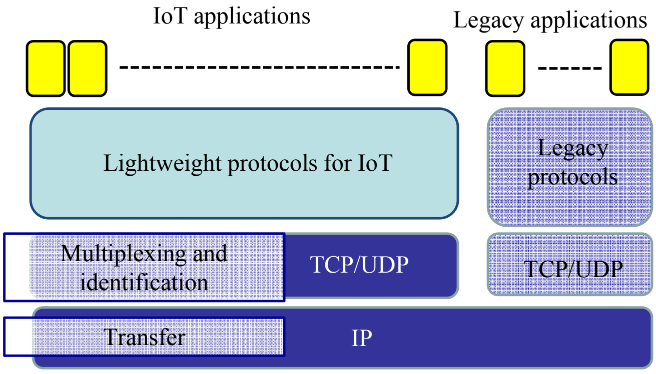

Fig. 3. Protocol stack including IoT communication

Regarding communication for IoT, functions of IP/TCP/UDP are minimized. Then, lightweight protocols for IoT are required over TCP/UDP. Meanwhile, for conventional communication, legacy protocols are invoked over TCP/UDP. HTTP is one of the promising protocols for Internet access.

III. HTTP for IoT Communication

It is assumed that HTTP is applied to communication for IoT. The HTTP must transfer a large number of tiny packets. Protocol overhead of HTTP causes serious problems, such as consumption of network resources and large delays.

Communication using HTTP is configured as shown in Figure 4. Sequence charts in this case are shown in Figure 5.

Since HTTP is operated over TCP/IP, reliable communication is provided. However, connections established by TCP are released on every access, because accessed data is transferred based on IP address and URL and their relationship is changed dynamically. In short, after many times of establishment of release of a connection, communication is completed. Therefore, communication for IoT causes serious overhead and consumption of network resources during this communication.

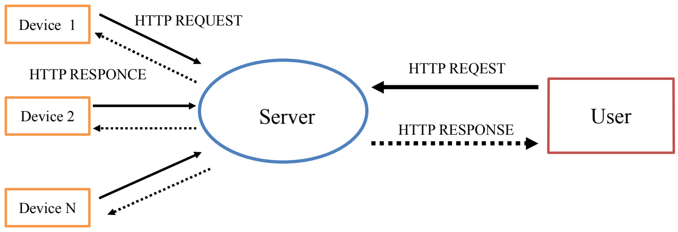

Fig. 4. System configuration using HTTP

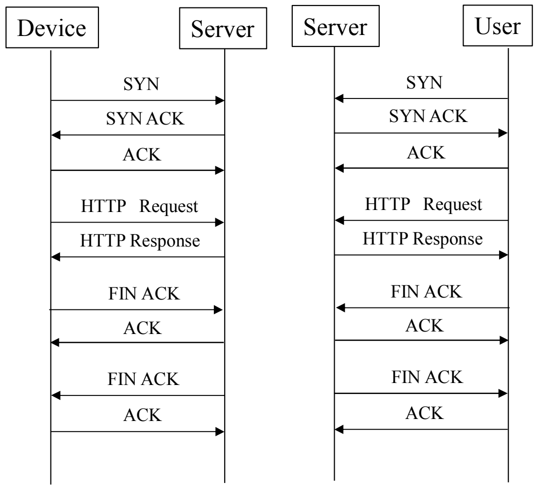

Fig. 5. Communication sequences on HTTP

IV. MQTT and its Performance

MQTT mitigates such protocol overheads in HTTP. This section describes sequences by MQTT for IoT communication.

A. Summary of operations in MQTT

MQTT provides three transfer modes based on the required reliability: QoS0 (Non assured transmission), QoS1 (Assured transmission), and QoS2 (Assured service on applications). QoS1 is similar to HTTP from a reliability point of view.

While HTTP is a symmetric protocol, MQTT has an asymmetric architecture for lightweight. Since, in most of the communication for IoT, non-intelligent distributed devices communicate with a server with intelligent ability, asymmetric communication is provided. Because of this point, MQTT is more suitable than HTTP.

MQTT consists of two message sets on a connection, "Publish" and "Subscribe." Data blocks are sent by Publish message and are received by Subscribe message. These data blocks are identified by "topic." Receiving data blocks are identified by topics registered by Subscribe message, in advance.

The system configuration is shown in Figure 6. In this configuration, communication sequence in monitor of devices by a user is shown in Figure 7. Communication sequences to control devices on MQTT are shown in Figure 8.

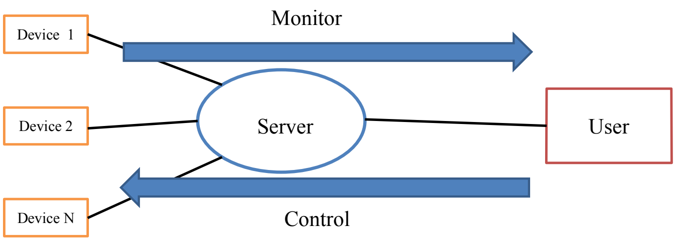

Fig. 6. System configuration using MQTT

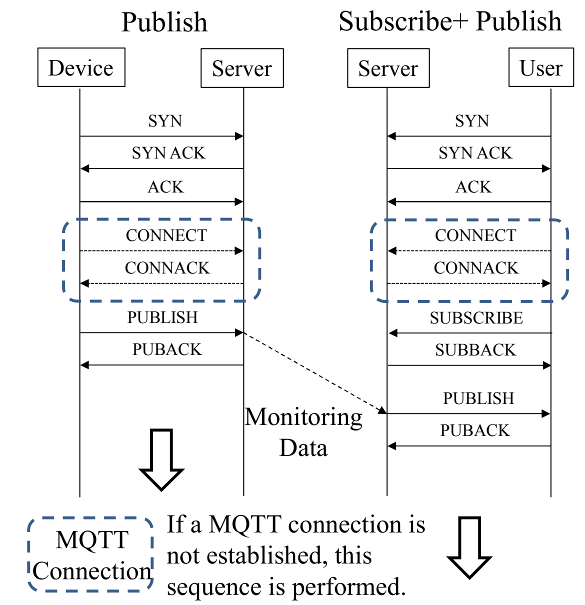

Fig. 7. Communication sequences in monior of a device on MQTT

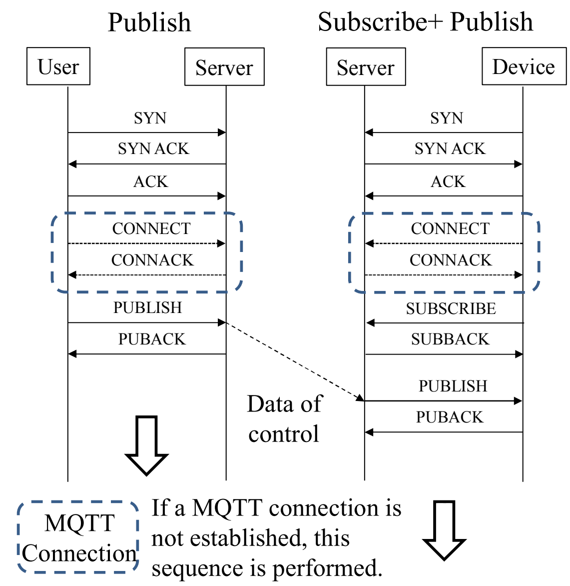

Fig. 8. Communication sequences to control a device on MQTT

B. Comparison of the required bandwidth for HTTP and MQTT

This sub-section compares the required bandwidth for HTTP and MQTT. Two comparison scenarios are described. One relates to the characteristics of the required bandwidth according to variable devices and the number of topics. Another relates to the characteristics of the required bandwidth according to variable data volume. These characteristics are shown in Figures 9 and 10.

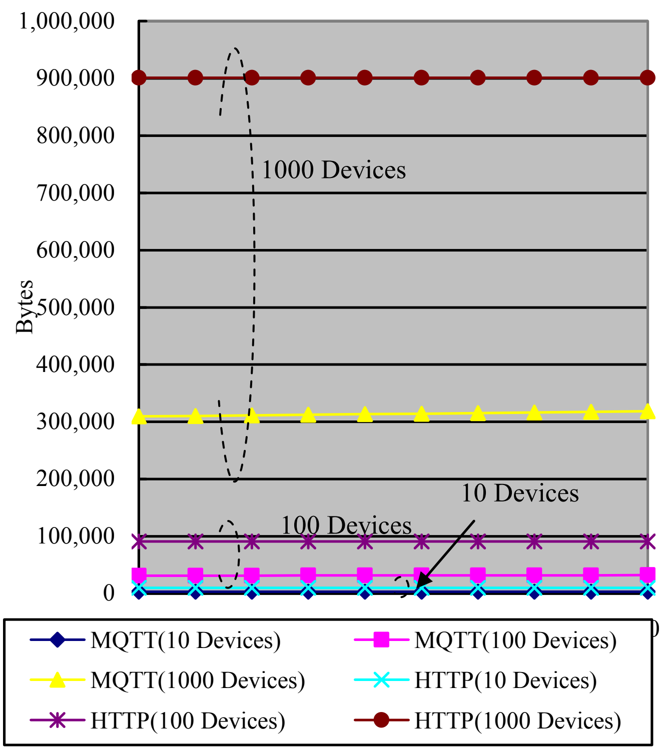

Fig. 9. Overhead by Topics in MQTT

Figure 9 shows the relationship between the total length of topics in MQTT and the number of transmission bytes. In this case, since payload size of applications is zero, the number of transmission bytes means only protocol overhead. Because HTTP does not have the concept of topics, the number of its transmission bytes depends on the horizon axis.

Figure 10 shows relationship between payload size and transmission bytes. In this case, the length of topics in MQTT is one byte.

These figures indicate that the number of transmission bytes depends on the number of devices connected in a server, as shown as in Figures 4 and 6.

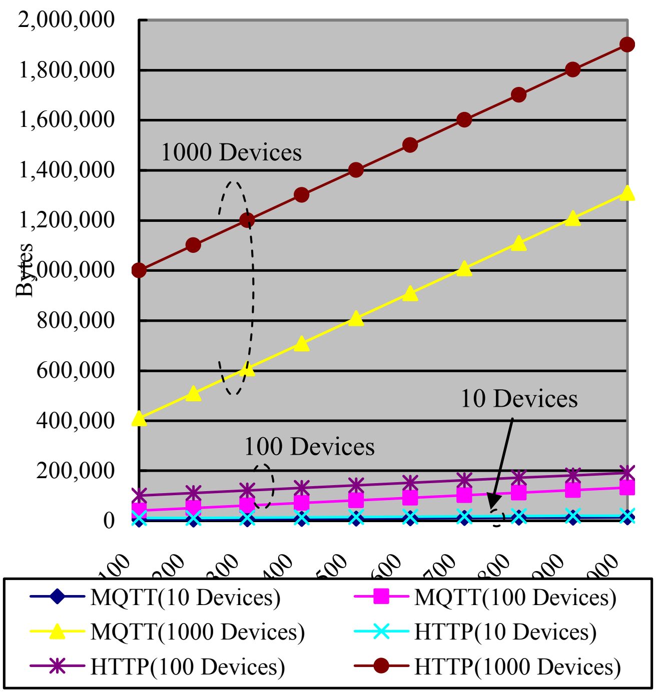

Fig. 10. Characteristics on payload size in HTTP and MQTT

These results imply that the protocol overhead of HTTP is larger than that of MQTT. In particular, if the number of connecting devices increases, as in typical IoT applications, this overhead is critical in HTTP.

C. Comparison of the resources in a server for HTTP and MQTT

This subsection discusses the required resources during communication in a server. Comparison with communication sequences from the first access to completion between MQTT and HTTP can be summarized, as shown in Table I.

TABLE I. COMPARISON WITH ROUND TRIPS IN SEQUENCES BETWEEN DEVICES AND A SERVER

		MQTT #1	MQTT #2	HTTP
No. of Round Trips		4	1	5
Required duration between start and completion	MAX	800ms	200ms	1s
	AVE	400ms	100ms	500ms

MQTT #1 includes establishment of MQTT connections. MQTT #2 assumes MQTT connections have been established in the previous communication.

If one way spends 100 ms across a network, the required duration between start of the communication and completion is described in Table I. 100 ms can be referred from QoS Class 0 of ITU-T Y.1541 [15] which has specified end-end performance over IP, as shown in Table II.

TABLE II. IP QOS REQUIREMENTS SPECIFIED IN ITU-T Y.1541

	QoS Classes					
	0	**1**	**2**	**3**	**4**	**5**
(1)	100 ms	400 ms	100 ms	400 ms	1 s	U
(2)	50 ms	50 ms	U	U	U	U
(3)	1×10^{-3}	1×10^{-3}	1×10^{-3}	1×10^{-3}	1×10^{-3}	U
(4)	1×10^{-4}					U

(1) IPTD: IP Transfer day
(2) IPDV: IP Delay Variation
(3) IPLR: IP Loss Rate
(4) IPER: IP Error Rate

This QoS class is most severe class and is applied to high quality VoIP. In this class, the specified maximum transfer day is 100 ms and delay variation is 50 ms, as shown in Table II.

During this period, a server has to assign its resources, such as buffers and processing power, to communication by HTTP or MQTT. To evaluate the required resources in a server, an M/M/∞//N queueing model [16] is assumed. In short, every communication is invoked by Poisson process. In the duration described in Table I, the average can be 50 ms considering delay variation (max 50 ms), and it complies with exponential distribution. Although the entire communication can be admitted by a server, the maximum number of communications is less than the number of devices.

Notations in this model are summarized as follows. The associated state transition diagram is shown in Figure 11.

λ= Arrival rate
h = average period of duration
k = the number of communication simultaneously
K= the number of devices

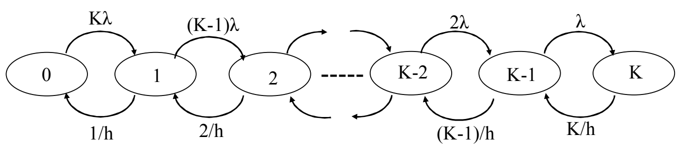

Fig. 11. State transition diagram in simultaneous communication

State probability that the number of communication is k, p_k, is derived in (1).

$$p_k = p_0 \prod_{i=0}^{k-1} \frac{\lambda(K-i)h}{i+1} = p_0(\lambda h)^k \binom{K}{k} \qquad (0 \le k \le K) \tag{1}$$

$$\binom{K}{k} \triangleq \frac{K!}{k!\,(K-k)!} \quad (2)$$

Therefore, p_k is derived by an explicit form as in (3). Then, the average of communication, i.e., the average of assigned resources, N is derived in (4).

$$p_k = \begin{cases} \frac{(\lambda h)^k \binom{K}{k}}{(1+\lambda h)^K} & (0 \le k \le K) \\ 0 & others \end{cases} \quad (3)$$

$$\mathrm{N} = \sum_{k=0}^{K} k \cdot p_k = \frac{\sum_{k=0}^{K} k \cdot (\lambda h)^k \binom{K}{k}}{(1+\lambda h)^K} = \frac{K \cdot \lambda h}{1+\lambda h} \quad (4)$$

Results of numerical calculations according to (4) are shown as in Figure 12. Figure 12 shows the case in which the number of devices is 100. Although the initial stage of MQTT (MQTT#1) is similar to HTTP, the stable stage of MQTT (MQTT#2) is much smaller than HTTP. As a result, MQTT provides light load for a server.

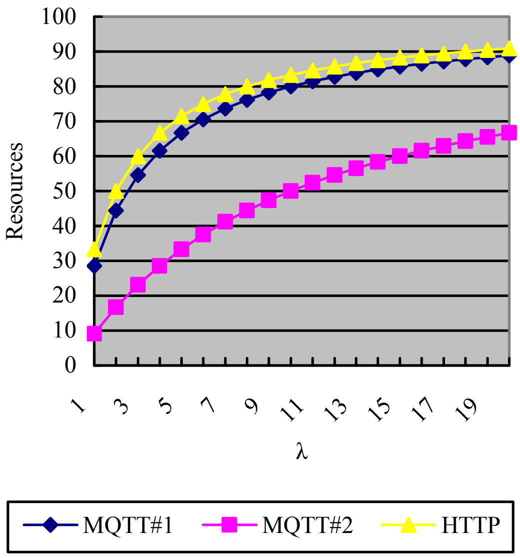

Fig. 12. Required server resources

V. ENHANCEMENT TO MQTT FOR REDACTION OF TRAFFIC

Advantages of MQTT have been described in the previous section. However, if the length of topics of MQTT increases, overhead of MQTT is relative large as shown in Figure 13, because topics of MQTT do not represent logical address or URL, but real information.

Figure 13 shows characteristics of the case that the number of devices is 10 and the payload size is zero. If the length of topics is more than 680 bytes, overhead of HTTP is smaller than that of MQTT.

To solve this problem, it is proposed that a server compresses topics registered by subscribe messages at the initial phase. Then, it advertises these compressed values to the devices and a user. When they send publish messages, the compressed value can be applied as shown in Figure 14. Figure 14 shows the case of monitoring data at a server. However, this proposal is applied to the case of control of data corresponding to Figure 8.

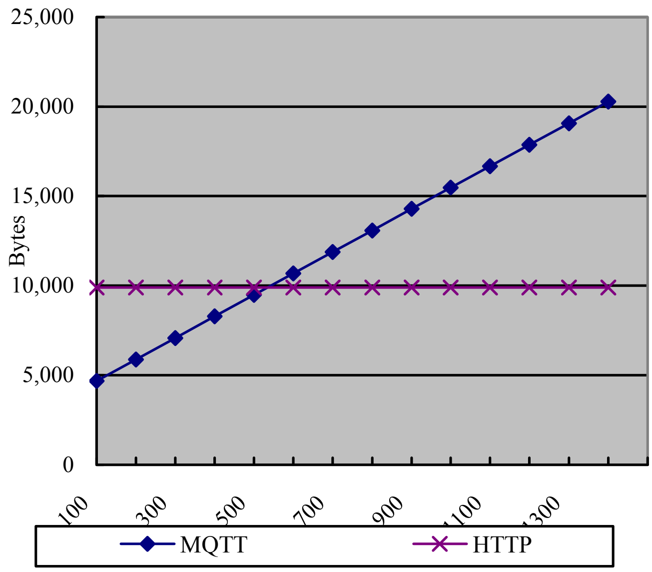

Fig. 13. Overhead in the case of long topics in MQTT

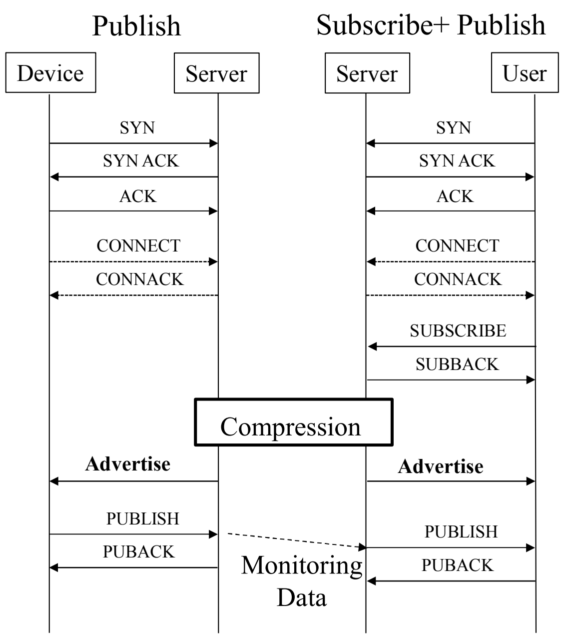

Fig. 14. Proosed mecnanism for enhancement of MQTT

Regarding implementation of compression in Figure 14, one of possible solutions is shown in Figure 15. In case of topics, “a/b/c/d”, memory blocks are allocated to each element. These blocks consist of “pointer” for chain of elements and, “data” for store of elements.

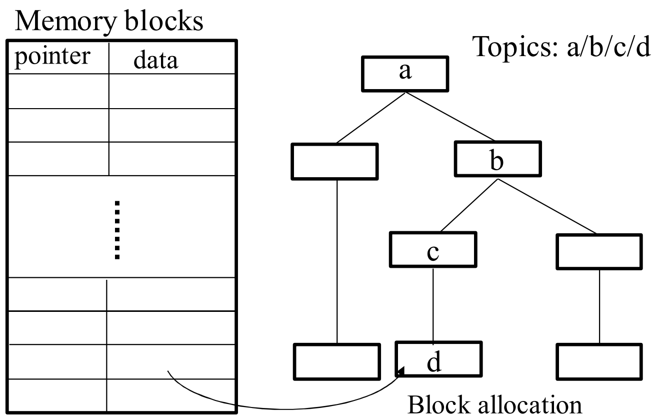

Figure 15 Implementation of compression using memory block allocation

VI. Conclusions

This paper has summarized the data transfer protocols used in IoT. IoT is expected to be applied to various applications as a social infrastructure. However, to deploy IoT widely, lightweight communication protocols are required. This paper has clarified that ICN architecture is the promising candidate for this purpose. A comparison has been made between the performance of HTTP in the category of legacy protocols and MQTT in the category of protocols based on ICN architecture. The paper concludes that MQTT performs better that HTTP. Additionally, this paper has proposed an approach to enhance MQTT.

Acknowledgements

A part of this research has been supported by Ministry of Economy, Trade and Industry (METI) project, “International standardization for the information communication equipment and device development based on IoT platform” We would like to express our gratitude to METI in Japan for support.

References

[1] T. Berners-Lee, R. Fielding, H. Frystyk, “Hypertext Transfer Protocol -- HTTP/1.0”, IETF RFC 1945, 1996

[2] M. Belshe, R. Peon, M. Thomson, “Hypertext Transfer Protocol Version 2 (HTTP/2)”, IETF RFC 7540, 2015

[3] J. Luo, C. Wu, Y. Jiang, J. Tong, “Name Label Switching Paradigm for Named Data Networking”, IEEE Communications Letters, Vol. 19, pp.335 - 338, 2015

[4] S. Eum, K. Nakauchi, Y. Shoji, N. Nishinaga, M. Murata, “CATT: Cache aware target identification for ICN”, IEEE Communications Magazine, Vol. 50, No.12, pp. 60 – 67, 2012

[5] C. Fang, R. Yu, T. Huang, J. Liu, Y. Liu, “A Survey of Green Information-Centric Networking: Research Issues and Challenges”, IEEE Communications Surveys & Tutorials, Vol. 17, No.3, pp. 1455 – 1472, 2015

[6] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, B. Ohlman, “A survey of information-centric networking”, IEEE Communications Magazine, Vo. 50, No.7, pp. 26 － 36, 2012

[7] G. Xylomenos, C. Ververidis, V. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. Katsaros, G. Polyzos, “A Survey of Information-Centric Networking Research”, IEEE Communications Surveys & Tutorials, Vol. 16, No.2, pp. 1024 – 1049, 2014

[8] M. Yamamoto, “Research trends on In-Network caching in content oriented networks”, IEICE Technical Report., Vol. 115, No.461, pp. 23 －28, 2015

[9] Md. F. Bari, S. Chowdhury, R. Ahmed, R. Boutaba, B. Mathieu, “A survey of naming and routing in information-centric networks”, IEEE Communications Magazine, Vol. 50, No.12, pp.43 －52, 2012

[10] M. Amadeo, C. Campolo, J. Quevedo, D. Corujo, A. Molinaro, A. Iera, R. Aguiar, A. Vasilakos, “Information-centric networking for the internet of things: challenges and opportunities”, IEEE Network, Vol. 30, No.2, pp. 92 - 100, 2016

[11] H. Yue, L. Guo, R. Li, H. Asaeda, “Yuguang Fang DataClouds: Enabling Community-Based Data-Centric Services Over the Internet of Things”, IEEE Internet of Things Journal, Vol. 1, pp. 472 － 482, 2014

[12] I. Sato, T. Kurira, K. Fukuda, T. Tsuda, “A extention of information centric for IoT applications”, Repeort of 3rd Technical committee on information centric networking (ICN), in IEICE, 2015

[13] IBM, “MQTT V3.1 Protocol Specification”, 2012, http://public.dhe.ibm.com/software/dw/webservices/ws-mqtt/mqtt-v3r1.html

[14] T. Fujita, Y. Goto, A. Koike, “M2M architecture trends and technical issues”, The Jurnal of IEICE, Vol.96, pp.305 － 312, 2013

[15] “Network performance objectives for IP-based services”, ITU-T Recommendation, Y.1541, 2011

[16] L. Kleinrock, “Queuing Systems, Volume 1: Theory”, Wiley Interscience, ISBN 0-471-49110-1, 1980