Analyses and Comparisons of Transmission Protocols for oneM2M Specification

Chi-Hua Chen[a]*, Ming-Shan Yeh[b], Yi-Hsin Lin[c], Chin-Chieh Huang[d], Shau-Sheng Tung[e], Wen-Hsien Chiang[f], Moa-Tai Chen[g], Kuen-Rong Lo[h]

Telecommunication Laboratories, Chunghwa Telecom Co., Ltd.,
No. 99, Dianyan Rd., Yangmei District, Taoyuan City 32661, Taiwan, R.O.C.

[a]chihua0826@cht.com.tw
[b]samiya@cht.com.tw
[c]weirdyi@cht.com.tw
[d]bdanny@cht.com.tw
[e]sstung@cht.com.tw
[f]kiang@cht.com.tw
[g]moutai@cht.com.tw
[h]lo@cht.com.tw

Abstract: Recently, Internet of Things (IoT) has become more and more popular. Several applications for IoT devices have been designed and developed to detect environmental changes and send instant updates to a cloud computing server farm via mobile communications and middleware for big data analyses. Multiple international organizations such as oneM2M have established specifications of M2M (Machine to Machine) communication and provided some use cases across multiple disciplines including energy, enterprise, healthcare, public services, residential, retail, transportation, among many others. Some middleware techniques which are defined by oneM2M include Representational State Transfer (REST), Constrained Application Protocol (CoAP), and Message Queue Telemetry Transport (MQTT) for transmission. However, the analyses and comparisons of these middleware techniques have not been discussed and investigated. Therefore, this study designed and implemented two transmission scenarios to compare the performance of each protocol. The experimental results showed that the response time of using MQTT protocol is the shortest. However, the transmission cost of using CoAP is the lowest. Therefore, the adaptable middleware technique can be selected and adopted in accordance with various IoT applications. Finally, two practical case studies including smart home and Intelligent Transportation System (ITS) in Taiwan are illustrated to discuss and to analyze the applications of these middleware techniques.

Keywords: Representational state transfer; constrained application protocol; message queue telemetry transport; internet of things; oneM2M

ISSN: 2222-7059 (print), ISSN: 2222-7067 (online), http://www.aspbs.com/aiem/
DOI: 10.7508/aiem.2015.02.001

1 INTRODUCTION

In recent years, Internet of Things (IoT) has become more and more popular [1]. The three layers in IoT are sensor, networking, and application layers [2-4]. For sensor and networking layers, the rise of mobile technology advancements (e.g., wireless sensor networking, smart mobile device, Long-Range Signaling and Control (LRSC), and Long Term Evolution (LTE) [5]) has led to a new wave of machine-to-machine (M2M), machine-to-human (M2H), human-to-human (H2H), and human-to-machine (H2M) communications [6]. For application layer, several mobile-based applications for mobile devices have been designed and developed to detect environmental changes and send instant updates to a cloud computing server farm via mobile communications and middleware for big data analyses. Multiple international organizations such as oneM2M [7, 8] and the European Telecommunications Standards Institute (ETSI) [9] have established specifications of M2M communication and provided some use cases across multiple disciplines including energy, enterprise, healthcare, public services, residential, retail, transportation, among many others. To illustrate the usefulness of such technologies, for instance, on-board units in cars can instantly detect and share information about the geolocation of the car, speed, following distance, and gaps with other neighboring cars. Such information can also be uploaded to a cloud computing server farm for obtaining the traffic information estimation to road users [10]. Although oneM2M defined some middleware techniques which include Representational State Transfer (REST) [11, 12], Constrained Application Protocol (CoAP) [13, 14], and Message Queue Telemetry Transport (MQTT) [15, 16] for transmission [8], the analyses and comparisons of these middleware techniques have not been discussed and investigated.

Therefore, this study designed and implemented two transmission scenarios to compare the performance of each protocol. The factors of response time and transmission cost which are the important characteristics of transmission protocol performance are analyzed and compared in the practical IoT environments. The advantages and limitations of these protocols are discussed for selecting the adaptable middleware technique in accordance with various IoT applications.

The remainder of this paper is organized as follows. In Section 2, this study illustrates the principles and fundamentals of each transmission protocols. The experimental results and case studies are presented in Section 3. Finally, this study concludes this paper in Section 4.

2 TRANSMISSION PROTOCOLS

This study discussed and focused on the transmission protocols (i.e., REST, CoAP and MQTT) which were defined in oneM2M specifications [8]. Necessary research background and relevant technology are presented as follows.

2.1 Representational State Transfer Protocol

REST protocol was proposed by Roy Fielding in 2000 to design a software architecture style for transmission [11]. The characteristics of REST protocol are client–server model, statelessness, cacheability, layered system, code on demand, and uniform interface. Moreover, REST protocol includes the following principles: (1) the states and functions of applications can be defined as resources; (2) an unique Uniform Resource Identifier (URI) can be used to identify a resource; (3) the uniform interfaces between client site and resource can be used to indicate the well-defined operations (i.e., GET, POST, PUT, and DELETE operations) and content types [17].

For transmission, three roles defined in REST protocol are data elements (e.g., resource, resource metadata, etc.), connectors (e.g., client, server, cache, etc.), and components (e.g., gateway, proxy, etc.) [11]. The client-server model and uniform interfaces can be provided based on these role responsibilities for simplified software requirements. Furthermore, the connectivity of REST protocol is stateless to reduce the usage of system resource and to improve the extendability.

Therefore, several web services have been designed based on the REST architecture style via Hypertext Transfer Protocol (HTTP) in recent years. Moreover, the standard data formats of JavaScript Object Notation (JSON) and eXtensible Markup Language (XML) are popularly adopted to develop the loosely coupled systems via REST protocol [18].

2.2 Constrained Application Protocol

CoAP was proposed and defined by the Internet Engineering Task Force (IETF) Constrained RESTful environments (CoRE) Working Group to build client-server model and the transmission between IoT and M2M devices via User Datagram Protocol (UDP) protocol [13]. Furthermore, CoAP makes the well-defined methods (i.e., GET, POST, PUT, and DELETE methods) [19] and supports several standard data formats (e.g., JSON, XML, Concise Binary Object Representation (CBOR), etc.) [20].

The important features of CoAP are low header overhead, low parsing complexity, and simple subscription for a resource. For instance, the length of CoAP header is only 4 bytes, and the CoAP Message Format (shown in Fig. 1) is illustrated as follows [13].

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Ver| T |  TKL  |      Code     |          Message ID           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Token (if any, TKL bytes) ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Options (if any) ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1 1 1 1 1 1 1 1|    Payload (if any) ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Fig. 1. CoAP message format [13]

- **Ver** is defined as the CoAP version.

- **T** is defined as the message type. Four message types in CoAP include Confirmable (CON), Non-Confirmable (NON), Acknowledgement (ACK), and Reset (RST).

- **TKL** is defined as the token length which is between 0 and 8 bytes for matching request messages.

- **Code** is defined the message type (i.e., request values: 1-31; responses values: 64-191; empty value: 0).

- **Message ID** is used to detect the redundant messages and to match the message for request and response.

- **Options** is defined some addition information (e.g., the format of payload, the value of entity tag, etc).

- **Payload** is defined as the main contents between devices.

2.3 Message Queue Telemetry Transport Protocol

MQTT protocol, a M2M and IoT connectivity protocol via Transmission Control Protocol (TCP), was originally proposed and developed by International Business Machines Corporation (IBM) and Eurotech [15]. The lightweight publish/subscribe messaging transport which is designed to build the connections between M2M and IoT devices with low network resources and hardware requirements includes three components as follows (shown in Fig. 2) [21, 22].

- **Topics** can be designed and created in a MQTT broker for publishing and subscribing.

- **Publishers** can publish their messages to a specific topic.

- **Subscribers** can subscribe to their interested topic. When messages are published to these topics, subscribers can receive the messages from MQTT broker.

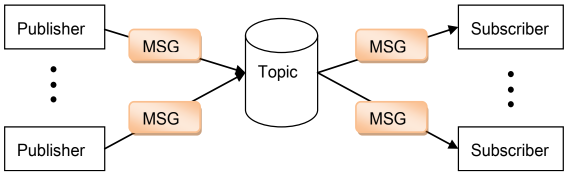

Fig. 2. MQTT publish/subscribe messaging transport [15]

The fixed header in MQTT packet includes MQTT control packet type (e.g., CONNECT, CONNACK, PUBLISH, SBUSCRIBE, UNSUBSCRIBE, DISCONNECT, etc.), flags specific to each MQTT control packet type (i.e., the kind of packet), and remaining length (shown in Fig. 3). Therefore, the length of this fixed header is only 2 bytes. Furthermore, the variable header is also supported in MQTT protocol to define the packet identifiers for various control packets.

Bit	7	6	5	4	3	2	1	0
Byte 1	MQTT Control Packet Type				Flags Specific to Each MQTT Control Packet Type			
Byte 2	Remaining Length							

Fig. 3. Fixed header format in MQTT protocol [15]

The advantages of MQTT protocol are summarized and discussed as follows [23].

- MQTT protocol can support one-to-one, one-to-many, many-to-one, and many-to many based on publish/subscribe messaging transport.

- The length of packet header is short for reducing the transmission cost.

- The Quality of Service (QoS) of transmission is supported for delivering messages between clients and servers. Three QoS levels defined in MQTT protocol include “at most once”, “at least once” and “exactly once”.

- Open source codes of MQTT protocol is published in Internet for development.

3 EXPERIMENTAL RESULTS AND CASE STUDIES

This section designed two transmission scenarios and illustrated two case studies to compare the performance of REST, CoAP, and MQTT protocols.

3.1 Experimental Results

Two senarios which are designed to evaluate the response time and transmission cost included Intranet and Internet (Shown in Fig.s 4 and 5). In each scenario, a server and a smartphone were implemented and transferred data via REST, CoAP, and MQTT protocols. The server provided a simple servcie and served as a REST server, a CoAP server, and MQTT broker. Moreover, the smartphone served as a REST client, CoAP client, and MQTT client (including publisher and subscriber). The smartphone can request server to serve the specific service via REST, CoAP, and MQTT protocols.

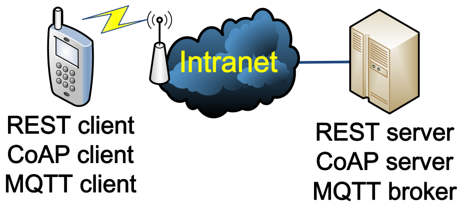

Fig. 4. Scenario 1: Intranet

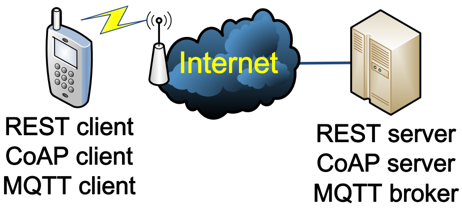

Fig. 5. Scenario 2: Internet

3.1.1 Response Time

For the analyses of response time, the experimental results indicated that MQTT protocol can provide the fastest response within 15 milliseconds (shown in Table 1). Moreover, CoAP required the most response time to serve. Therefore, MQTT protocol is suitable for the instantaneous services.

Table 1. The comparisons of Response Time (Unit: milliseconds)

Scenario	MQTT	REST	CoAP
Scenario 1 (Intranet)	5	14	15
Scenario 2 (Internet)	15	31	47

3.1.2 Transmission Cost

For the analyses of transmission cost, Table 2 showed that CoAP based on UDP required the lowest transmission cost about 136 bytes. However, MQTT and REST are based on TCP, so more transmission cost is required for TCP three-way-handshake. Therefore, CoAP is suitable for the services with mass data transmission.

3.2 Case Studies

In this study, two practical case studies including smart home and Intelligent Transportation System (ITS) in Taiwan were selected to discuss and to analyze the applications of these middleware techniques.

Table 2. The comparisons of Transmission Cost (Unit: Bytes)

Item	MQTT	REST	CoAP
TCP Three-way-handshake	186	186	0
Connection	143	0	0
Subscription	133	0	0
Request and Publishing	84	133	71
Response and Pushing	78	259	65
Disconnection and Guaranteed Delivery	54	228	0
Summary	**678**	**806**	**136**

3.2.1 Smart Home

In the case study of smart home, the customer requirement is to instantaneously monitor home environmental information for security improvement. The architecture includes sensors, smart hub, and server farm (shown in Fig. 6). The sensors can detect the environmental information and transfer this information to smart hub. For instantaneous monitor, the smart hub transfers this information to server farm via MQTT protocol with lower response time.

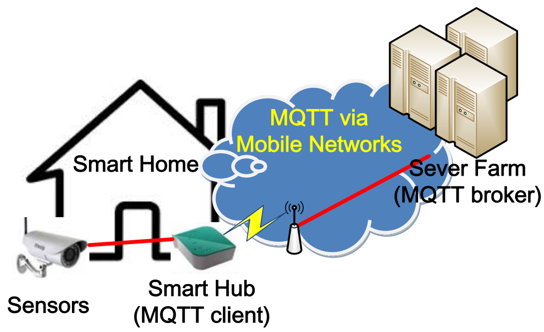

Fig. 6. Smart home architecture

3.2.2 Intelligent Transportation System

In the case study of ITS, the Commercial Vehicle Operation Service (CVOS) requirement is to trace the location and status of vehicle. The architecture includes On-Board Unit (OBU), server farm, and web-based clients (shown in Fig. 7). OBU can detect the location and status of vehicle and send to server farm. Due to the requirement of mass vehicle information transmission, CoAP is considered and adopted to build the connection between OBU and server farm for reducing the transmission cost.

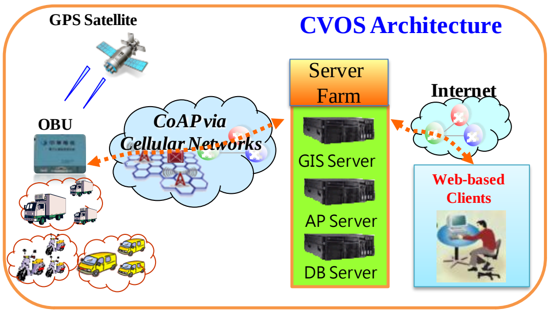

Fig. 7. CVOS system architecture

4 CONCLUSIONS AND FUTURE WORK

This study designed and implemented two transmission scenarios to compare the performance of each protocol. The experimental results showed that the response time of using MQTT protocol is the shortest. However, the transmission cost of using CoAP is the lowest. Therefore, the adaptable middleware technique can be selected and adopted in accordance with various IoT applications. In the future, the various M2M and IoT applications can refer the experimental results in this study to adopt the adaptable middleware technique for the implementation of their applications.

References

[1] D. Guinard, V. Trifa, S. Karnouskos, P. Spiess, D. Savio, Interacting with the SOA-based internet of things: discovery, query, selection, and on-demand provisioning of web services, IEEE Transactions on Services Computing, vol. 3, no. 3, pp. 223-235, 2010.

[2] G. Moritz, F. Golatowski, C. Lerche, D. Timmermann, Beyond 6LoWPAN: web services in wireless sensor networks, IEEE Transactions on Industrial Informatics, vol. 9, no. 4, pp. 1795-1805, 2013.

[3] W. He, L.D. Xu, Integration of distributed enterprise applications: a survey, IEEE Transactions on Industrial Informatics, vol. 10, no. 1, pp. 35-42, 2014.

[4] L.D. Xu, W. He, S. Li, Internet of things in industries: a survey, IEEE Transactions on Industrial Informatics, vol. 10, no. 4, pp. 2233-2243, 2014.

[5] H.S. Dhillon, H.C. Huang, H. Viswanathan, R.A. Valenzuela, Power-efficient system design for cellular-based machine-to-machine communications, IEEE Transactions on Wireless Communications, vol. 12, no. 11, pp. 5740-5753, 2013.

[6] A. Ksentini, Y. Hadjadj-Aoul, T. Taleb, Cellular-based machine-to-machine: overload control, IEEE Network, vol. 26, no. 6, pp. 54-60, 2012.

[7] J. Swetina, G. Lu, P. Jacobs, F. Ennesser, J.S. Song, Toward a standardized common M2M service layer platform: Introduction to oneM2M, IEEE Wireless Communications, vol. 21, no. 3, pp. 20-26, 2014.

[8] oneM2M, Functional Architecture, TS-0001-V1.6.1, oneM2M Partners, 2015.

[9] O. Del Rio Herrero, R. De Gaudenzi, High efficiency satellite multiple access scheme for machine-to-machine communications, IEEE Transactions on Aerospace and Electronic Systems, vol. 48, no. 4, pp. 2961-2989, 2012.

[10] C.H. Chen, H.C. Chang, C.Y. Su, C.C. Lo, H.F. Lin, Traffic speed estimation based on normal location updates and call arrivals from cellular networks, Simulation Modelling Practice and Theory, vol. 35, no. 1, pp. 26-33, 2013.

[11] L. Richardson, S. Ruby, RESTful Web Services, O'Reilly Media, California, USA, 2007.

[12] J.I. Fernandez-Villamor, C.A. Iglesias, M. Garijo, A framework for goal-oriented discovery of resources in the RESTful architecture, IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 44, no. 6, pp. 796-803, 2014.

[13] Z. Shelby, K. Hartke, C. Bormann, The Constrained Application Protocol, Request for Comments: 7252, Internet Engineering Task Force, 2014.

[14] S.N. Han, G.M. Lee, N. Crespi, Semantic context-aware service composition for building automation system, IEEE Transactions on Industrial Informatics, vol. 10, no. 1, pp. 752-761, 2014.

[15] G. Brown, L.P. Lamoureux, MQTT and the NIST Cybersecurity Framework, Version 1.0., OASIS Open, 2014.

[16] A.M. Zambrano, I. Perez, C. Palau, M. Esteve, Distributed sensor system for earthquake early warning based on the massive use of low cost accelerometers, IEEE Latin America Transactions, vol. 13, no. 1, pp. 291-298, 2015.

[17] S.P. Onga, S. Choliab, A. Jainb, M. Brafmanb, D. Gunterb, G. Cederc, K.A. Persson, The materials application programming interface (API): a simple, flexible and efficient API for materials data based on REpresentational State Transfer (REST) principles, Computational Materials Science, vol. 97, pp. 209-215, 2015.

[18] R.D. Sadafule, Mobile app development for the Indian market, IEEE Software, vol. 31, no. 3, pp. 17-20, 2014.

[19] P. Madhumitha, B. Johnsema, D. Manivannan, Domination of constrained application protocol: a requirement approach for optimization of Internet of things in wireless sensor networks, Indian Journal of Science & Technology, vol. 7, no. 3, pp. 296-300, 2014.

[20] D. Chen, M. Nixon, S. Han, A.K. Mok, X. Zhu, WirelessHART and IEEE 802.15.4e, Proceedings of 2014 IEEE International Conference on Industrial Technology (ICIT), Busan, Korea, 2014.

[21] F. Palumbo, J. Ullberg, A. Štimec, F. Furfari, L. Karlsson, S. Coradeschi, Sensor network infrastructure for a home care monitoring system, Sensors, vol. 14, no. 3, pp. 3833-3860, 2014.

[22] J. Park, M.J. Lee, SCondi: a smart context distribution framework based on a messaging service for the Internet of things, Journal of Applied Mathematics, vol. 2014, Article ID 271817, 8 pages, 2014.

[23] S.T.B. Hamida, E.B. Hamida, B. Ahmed, A new mHealth communication framework for use in wearable WBANs and mobile technologies, Sensors, vol. 15, no. 2, pp. 3379-3408, 2015.