oneM2M : Extension of Protocol Binding

Reuse of Binding Protocol's Legacy Services

Jun Hwan Huh
Dept. of Computer Science and Engineering
Pusan National University
Busan, South Korea
ijhhuh@pusan.ac.kr

Dong Hyun kim
Dept. of Computer Science and Engineering
Pusan National University
Busan, South Korea
dhkim1106@pusan.ac.kr

Jong Deok Kim*
Dept. of Computer Science and Engineering
Pusan National University
Busan, South Korea
kimjd@pusan.ac.kr

***Abstract*— The oneM2M is a partnership project aimed at developing the technical specification of global service platform for the Internet of Things (IoT). The project deals with various parts, but WG3(: WG PRO)in particular, defines the exchanging message protocol between the entities(oneM2M Primitive), oneM2M core protocol to handle errors and bindings between core protocol and transport layer protocol(CoAP, HTTP, MQTT). The protocol binding is when one or more than one interfaces are combined with other protocols. For the protocol binding, convergence protocol is necessary which converts the interface between the other protocols. This study handles the protocol convergence interface that is defined in binding protocol techniques published by the standard development organization called oneM2M. Legacy "protocol binding" specification is focused on message translation between oneM2M's request/response and binding target protocol's message, but this paper suggests about binding services between oneM2M and target binding protocol defined. The extension of this study considers service mapping between protocols and the expected benefits from the extended one with examples of each benefit.**

Keywords — oneM2M; Protocol Binding; IoT; M2M;

I. INTRODUCTION

The Internet of Things (IoT) is the technique and service that enable physical objects to communicate and exchange data. The term of the Internet of Things or IoT in short was firstly coined by Kevin Ashton in 1999 while working at the MIT-Auto-ID centers in 1999. It was when the horizontal structured architecture was getting attention to overcome the weakness of the vertical service structures of the past. The key concept of the horizontal structure, middleware/platform was also starting to serve as a significant role. The middleware/platform having various interfaces was presented by lots of nations and entrepreneurs imprudently, in turn this phenomena brought up the complex interoperability between middleware/platforms.

The oneM2M organization defined the standard in the perspective of application/service regarding the common service standard in M2M/IoT. The organization defined 12 services in the Common Service Entity (CSE) and each service is provided by the Common Service Function (CSF). Those 12 services that the current CSF has been device management, group management, register, discovery, subscription and notification, application/service management, security, communication management and charging management.

One of the primary characteristics of the oneM2M is resource based architecture. Every service, function and data are expressed in terms of resources and each resource is a tree structure with every attribute specified in the oneM2M standard.

Every resource was designed based on RESTful architecture. The CSFs provides services by CREATE, RETE, UPDATE, DELETE, NOTIFY(: CRUDN) method controls toward the resources. The CSFs offer services through API reference points that are called Mcc(CSE to CSE), Mca(CSE to AE), Mcn(CSE to NSE). The request and reply about each API is represented by the Primitive(oneM2M: Request/response) that the oneM2M has defined.

The oneM2M document, TS-008, 009, 010, specifies the protocol binding standard between CoAP, HTTP and MQTT, which means that the binding is to combine one or more than one interface between other protocols.

This study illustrates the extended form of protocol binding approach that is considered not only message mapping but service mapping between protocols. Accordingly, the extended forms of primitive and binding function are presented in this study.

II. PROTOCOL BINDING

The protocol binding allows one or more than interfaces to be combined between other protocols. The expected benefits of the protocol binding to the middleware/platform are as follows:

- Extendibility.
  It increases the horizontal connection by using various protocols.
- Simplicity
  The elements that architecture should consider becoming simple while supporting the existing protocol interface.
- Adaptability
  The architecture is able to adapt to the environments of the legacy protocol by using the resource/service of them.
- Reusability
  The resource/service of the legacy protocol is available, its new definition is not needed.

oneM2M : Primitive

The Primitive is composed of control part and content part.

Control part, which contains parameters specifying the processing of the primitive and content part, which represents resource data[2].

A request/response about the resource is transmitted via primitive and oneM2M specification defined 5 kinds of operation which is CRUDN.

oneM2M : Protocol Binding

To implement the protocol binding, the oneM2M illustrates the interface conversion standard between the oneM2M request/response primitive and the target protocol. The oneM2M documents, TS-008(CoAP Protocol Binding) ,009(HTTP Protocol Binding) ,010(MQTT Protocol Binding), cover message conversion standard between the oneM2M request/response primitive and the underlying networks(CoAP, HTTP, MQTT).

The binding function allows the primitive request/response messages to be converted into the message of the target protocol, so that entity is able to use the underlying network protocols.

* : Corresponding Author

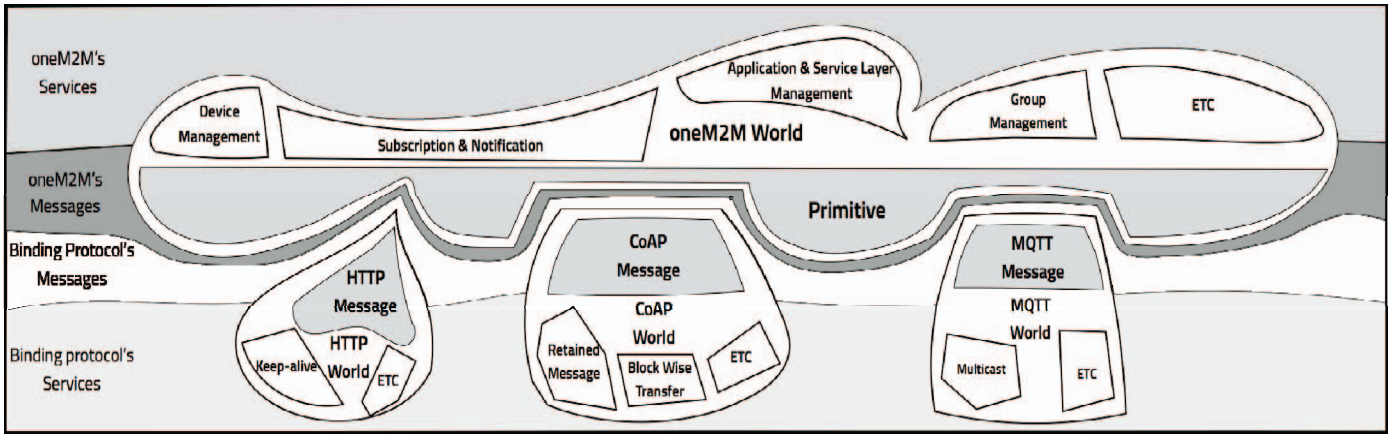

Figure 1 Protocol Binding oneM2M Primitive with Underlying Network Protocols

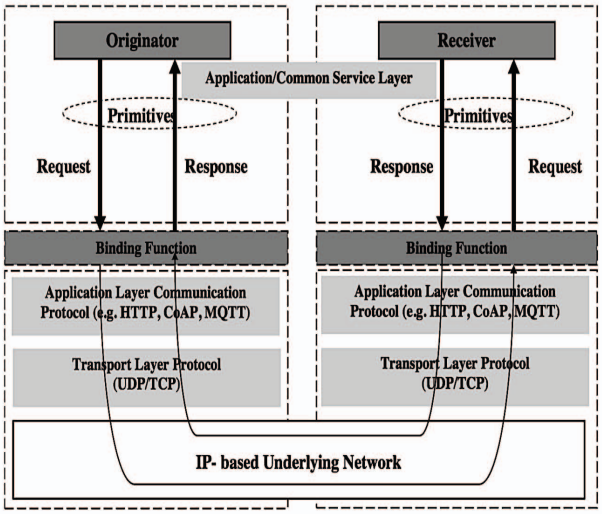

Figure 2 Communication model using Request and Response primitives over an Underlying Network [2]

III. Method of Approach

The current protocol binding method of the oneM2M focused on the mapping between primitive attribute and binding target protocol options. Since the current binding function is focused on the conversion of primitives, it is not possible to reference oneM2M's service attribute. Figure 3 illustrates the extension of service mapping between the oneM2M service and the target protocol. The point is reusing the similar services between other protocols and the benefits are as follows.

- Reduce implementation complexity
- Increase service efficiency
  - Reduce transmission amount of message
  - Reduce transmission time of message

IV. Suggestion

In the extension process from protocol binding service to service mapping, the approaches are divided into two parts. Firstly, the simple message mapping approach is focused on mapping with message attributes of the target protocol aimed at binding between the control and contents part of the oneM2M primitive. Secondly, the service mapping approach is to map services of CSFs of the oneM2M and the target protocol (refer to Figure4). In other words, this

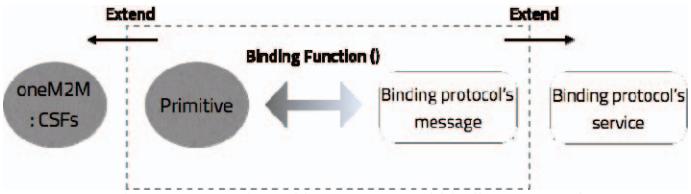

Figure 3 Extension of legacy protocol binding approach

approach is mapping message options regarding CSF related resources and service of the target protocol. To implement these two methods, the primitive and binding functions are extended.

The protocol binding method suggested by the oneM2M is mapping attributes of primitive and options of the target protocol. The problem of this method is that it is not possible to consider the attributes of the oneM2M service in binding function. Therefore, this study suggests the needs of the service instructor who informs the primitive related services of the oneM2M. The instructor adds to the control part of the primitive, so that the instructor helps mapping to the target protocol's service while considering the resources of oneM2M in the binding function (Refer to Figure4). Accordingly, binding function is needed to be extended to allow for the service instructor.

V. Extension of Protocol BInding With Example

This chapter gives examples of the benefits from the extension of protocol binding in the oneM2M.

First, Reduce implementation complexity

Software management service (: ASM CSF) of the oneM2M is in charge of managing software. For the firmware update, CoAP is assumed to be used. When using limited size of CoAP message based on the current UDP is used to transmit the large size of the firmware, the management for the sequence and combination of message is required. But using the block wise transfer from CoAP reduces the implementation complexity because of block wise transfer already consideration of the sequence and combination of message Issue.

Second, Increase service efficiency

The Group management service of the oneM2M (: GMG CSF) is in charge of managing group resources. One of attribute, fanOutPoint is a virtual attribute that is able to spread the transmitted message to group lists. Similarly, there is a multicast service to transmit CoAP messages to the majority of people.

The author gives examples of reducing transmission amount of message by mapping. Figure 5 illustrates when the current legacy protocol binding approach is used. ADN-AE(: Application Entity) tries to acquire several CSE resources through IN-CSE. Then, the

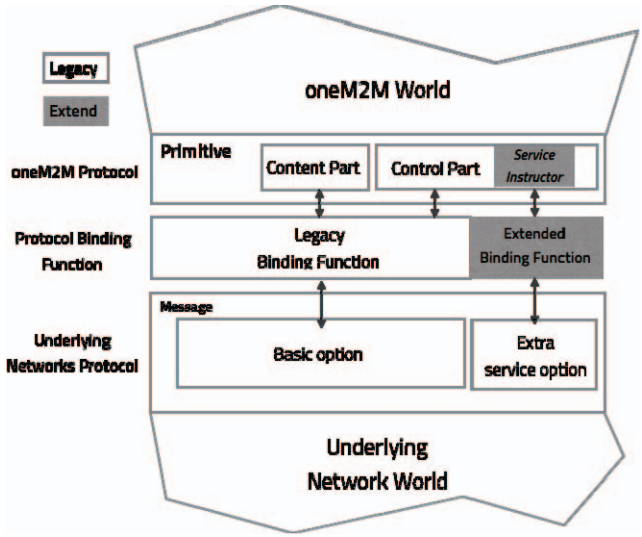

Figure 4 Extension of primitive & binding function

message refers to the group resources and is transmitted to those CSE. The numbers of the messages transmitted match the one of CSEs. fannedOutPoint and Multicast of CoAP then transmits them to the group, which sends multicast messages once and reduces the transmission amounts of messages.

The notification service of the oneM2M(: SUB CSF) is to notify any changed information according to the changes of resources/attributes. One of the attributes, preSubscriptionNotify, is a kind of caching services that transmit previously published data to new subscribers. Similar to this attribute, there is a retained message service of the MQTT, which serves the same role. The example of reducing transmission time by mapping the preSubscriptionNotify and retained message service is as follows. Figure 7 depicts the situation when the legacy protocol approach is used. The IN-CSE serves a role of a publisher that informs subscribers. The ADN-AE receives previously published information when subscription. When the IN-CSE is requested to subscribe, sends previously published information to ADN-AE. Figure 8 shows when the extended protocol binding approach is used. Even if the CSE did not receive the inquiry from binding the preSubscriptionNotify attributes and retained message services, the MQTT server will provide the same services. Instead of the CSE, MQTT server performs the service; therefore, the transmission time of the message is reduced.

I. Conclusion

This study suggested the need for the extension of legacy protocol binding approach to achieve the mapping between oneM2M service and binding target protocol. The study was based on the protocol binding related technical documents published by the oneM2M. Then, the author suggested the benefits of this extension and gave examples of each benefit. To implement the extension approach the suggestions in this study is to research about services that each underlying network provides, added values that is resulted from mapping with the service of the oneM2M. Lastly, the structure of the oneM2M should be defined in detail.

II. Acknowledgement

This work was supported by the ICT R&D program of MSIP/IITP. [2015(1711021663) Research on Implementation Techniques of Realistic Experience-type Contents based Smart Street]

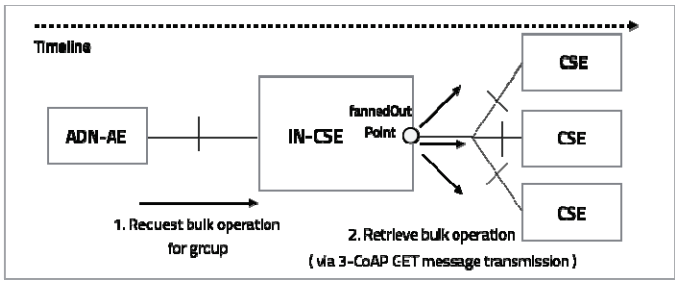

Figure 5 Legacy protocol binding approach with GMG CSF

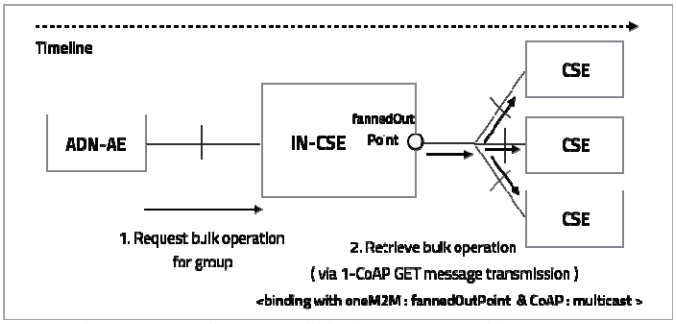

Figure 6 Extension protocol binding approach with GMG CSF

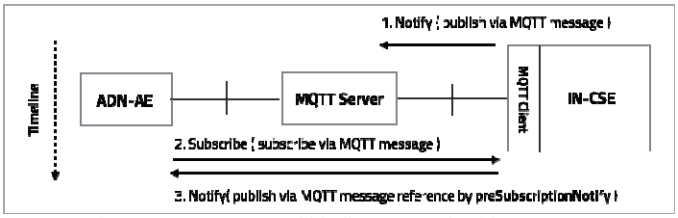

Figure 7 Legacy protocol binding approach with SUB CSF

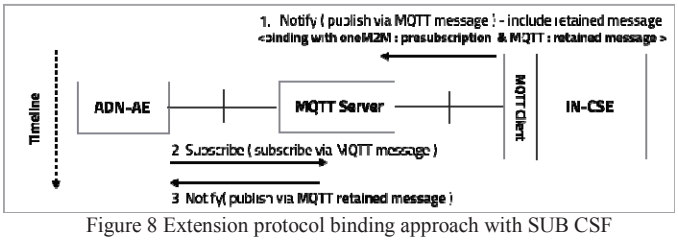

Figure 8 Extension protocol binding approach with SUB CSF

References

[1] oneM2M-TS-0001, “Functional Architecture”, v1.6.1, January, 2015.

[2] oneM2M-TS-0004, “Service Layer Core Protocol Specification”, v1.0.1, January, 2015.

[3] oneM2M-TS-0008, “CoAP Protocol Binding”, v1.0.1, January, 2015.

[4] oneM2M-TS-0009, “HTTP Protocol Binding”, v1.0.1, January, 2015.

[5] oneM2M-TS-0010, “MQTT Protocol Binding”, v1.0.1, January, 2015.

[6] Shelby, Zach, and Carsten Bormann. "Blockwise transfers in CoAP." (2014).

[7] Banks, A., and R. Gupta. "MQTT Version 3.1. 1." OASIS Standard (2014)