Thesis for the Degree of Master of Science

Design and Implementation of CoAP-Based Messaging System for IoT Healthcare Services

School of Electronics Engineering,
Major in Information and Communication Engineering
The Graduate School

Kim Hyunsu

June 2016

The Graduate School
Kyungpook National University

Design and Implementation of CoAP-Based Messaging System for IoT Healthcare Services

Kim Hyunsu

School of Electronics Engineering,
Major in Information and Communication Engineering
The Graduate School

Supervised by Professor Park Jongtae

Approved as a qualified thesis of Kim Hyunsu
for the degree of Master of Science
by the Evaluation committee

June 2016

Chairman ________ ㉤

________ ㉤

________ ㉤

The Graduate School
Kyungpook National University

Contents

I. Introduction

1. Motivation

Recently, due to the advancement of Internet of Things, it is being implemented in various industries such as healthcare, smart city, smart factory, smart farm, etc. Above all, IoT healthcare service has been focused on globally in the healthcare industry.

To provide IoT healthcare services, various studies are continuously being done on converging IoT and healthcare technology. One of the studies that has been done is the convergence of oneM2M and ISO/IEEE 11073 PHD standards. The use of international standards can help to solve problems on the existing vertical structure such as costs of system installation, extension and maintenance, and time-consumption during development.

oneM2M provides a set of standards to provide a horizontal platform architecture, enabling applications to connect securely through standardized APIs. oneM2M incorporates the most commonly used industry protocols for IoT, such as MQTT, CoAP and HTTP [1].

ISO/IEEE 11073 PHD(Personal Health Devices) is one of the standards for healthcare and it defines how to send, monitor and control biometric information using a healthcare device [2].

ISO/IEEE 11073 PHD standard is an independent transport protocol, thus it's implementation is shared between different devices and different wireless or wired transport technologies (Bluetooth, Zigbee, USB). It does not consider the internet protocols in the under layer communication protocol. Therefore, to provide IoT healthcare services, it requires for ISO/IEEE 11073 PHD to be integrated with oneM2M internet protocols.

To prevent the above described problems for IoT healthcare service, various research studies have been done. In [3], the ISO/IEEE 11073 communication model was adapted to the MQTT protocol, and thus enabling publish/subscribe model. Other works have been done to integrate CoAP protocol with ISO/IEEE 11073 PHD, such as [4], [5].

However, [3] has the advantages of supporting many to many communications, Quality of Service and small overhead, but it has a disadvantage of limiting the rest time of devices because it always maintains a connection for data transfer. [4] has the limitation of only providing one-to-one communication for 11073

Event Report model. In [5], has the limitation of constrained PHD's rest time, because it used CoAP server on the PHD so PHD always keeps on waiting to receive requests.

In order to solve the drawbacks that other protocols offer in IoT healthcare, in this paper, a design and implementation of Publish/Subscribe messaging system for IoT healthcare services using CoAP has been developed.

The proposed system is designed by combining the advantages of CoAP and MQTT and these are small message size, QoS, etc and supports many to many communication in IEEE 11073 Event Report respectively.

2. Research Originality and Contribution

Compared with other related works, the originality of the proposed approach is as follows:

- For the provision of IoT healthcare service, we proposed a new architecture by converging oneM2M standards and ISO/IEEE 11073 PHD standards.
- In [5], the Event Report of ISO/IEEE 11073 PHD was provided by using CoAP observe method. However, this

structure has limitations in providing many-to-many communication model. To overcome the limitations, in this paper, we can provide not only existing one-to-many and many-to-one communication models but also cater for many-to-many communication model by using publish/subscribe structure that is based on CoAP.

The contribution of our research is as follows:

- The architecture and signaling procedure of Publish/Subscribe messaging system using CoAP was designed to provide IoT healthcare service.
- Implements the prototype of publish/subscribe messaging system using CoAP.
- For evaluation of proposed system messaging System, we compared its performance with the existing MQTT messaging system.

3. Thesis Organization

The remainder of this thesis is organized as follows, In Section Ⅱ, we describe the IoT healthcare architecture, healthcare standards and application layer protocols for constrained IoT environment. In Section Ⅲ and Ⅳ, we describe the architecture, implementation and results of the proposed system and finally, conclude on the research.

II. Related Work

In this chapter, we introduce the architecture of CHA architecture for providing healthcare services and then explain the international healthcare standards. We then give an introduction to application layer protocols used for communication in a constrained IoT environment.

1. Continua Health Alliance

Philips was one of the founding members of the Continua Health Alliance(CHA) in 2006. CHA is an organization that is comprised of technology, medical device and healthcare industry leaders dedicated to making personal healthcare a reality[6]. Currently, the Alliance consists of over 260 member companies all working together towards the same goal of improving healthcare worldwide by establishing a system of interoperable personal healthcare solutions that fosters independence and empowers people and organizations to better manage health and wellness[7].

To implement healthcare services, there is a need to define end-to-end architecture; from personal health device for health data measurement to the analysis by the medical system.

To solve the mentioned problem, CHA publishes the Continua Design Guidelines, which provide a flexible implementation framework for authentic interoperability of personal connected health devices and systems. They define the interfaces that enable the secure flow of medical data among sensors, gateways, and end services, removing ambiguity in underlying healthcare standards and ensuring consistent implementation through product certification.

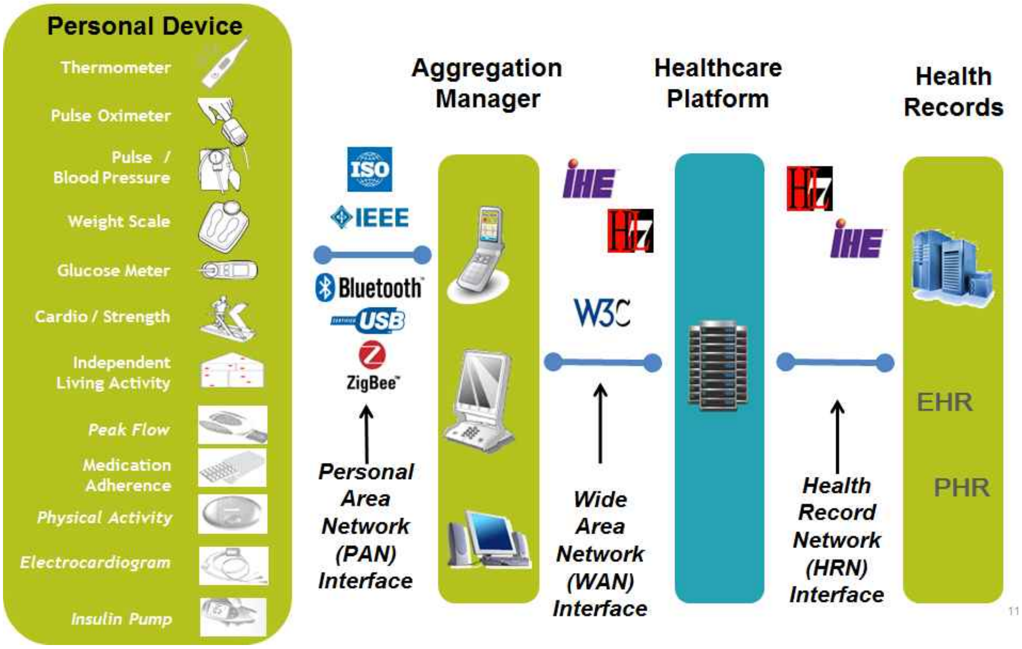

Figure 1 Continua Health Alliance end-to-end architecture

1.1 Continua Design Guideline in IoT

Figure 2 shows an enhanced Continua design guidelines with IoT healthcare service structure. In IoT environment, the wearable device can be substituted for personal health device, and the devices can connect to the internet using IP network. This characteristic was not considered in the continua design guideline. For this reason, many kinds of research are being done on healthcare data transfer using internet protocols such as HTTP, MQTT, CoAP [3][4][5].

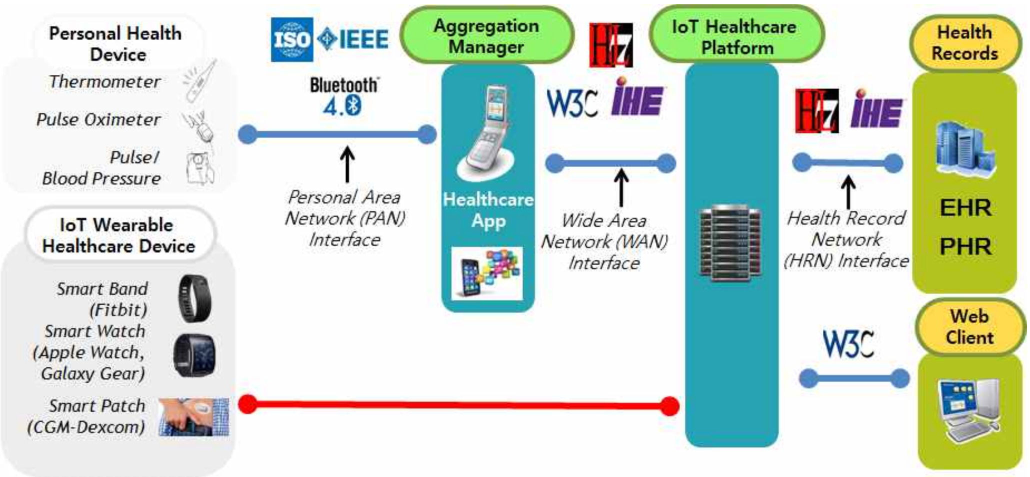

Figure 2 Continua Health Alliance end-to-end architecture in IoT environment

In IoT healthcare environment, the transaction of the measured healthcare data from healthcare devices can be performed in two ways; Using internet connection or locally.

Using local transmission, healthcare devices can transact using Bluetooth, ZigBee, and USB, etc that are based on ISO/IEEE 11073 PHD standard. For internet connection, internet protocols like HTTP, MQTT, CoAP, etc are used.

2. Healthcare Standards

For providing healthcare services, interoperability of each device is important. To do this, many international standard organizations have standardized healthcare services, such as ISO/IEEE 11073 PHD(Personal Health Devices) by ISO and IEEE, HL7 CDA(Clinical Document Architecture) by HL7 and PCD(Patient Care Device) by IHE.

Health Level Seven International (HL7) is a not-for-profit, ANSI-accredited standards developing organization dedicated to providing a comprehensive framework and related standards for the exchange, integration, sharing, and retrieval of electronic health information that supports clinical practice and the management, delivery and evaluation of health services [8].

The main goals of HL7 are to provide standards for exchanging healthcare data between computer healthcare applications and to provide interoperability for healthcare information systems. Due to this, many healthcare service providers have used HL7 for coordinating each independent systems.

Integrating the Healthcare Enterprise (IHE) is an international initiative to promote the use of standards to achieve interoperability among health information technology (HIT) systems and effective use of electronic health records (EHRs).

The vison of IHE PCD is the nexus for vendors and providers to jointly define and demonstrate unambiguous interoperability specifications, called profiles, which are based on industry standards, and which can be brought to market [9].

2.1 ISO/IEEE 11073 PHD

ISO/IEEE 11073 PHD standard is for providing interoperability between personal health devices and health manager. ISO/IEEE 11073 PHD consists of ISO/IEEE 11073-20601 and IEEE 11073-104xx. ISO/IEEE 11073 PHD defines OSI 5-7 layers and is independent of transport protocols [10]. Due to this independence of the transport protocols, ISO/IEEE 11073 PHD can use Bluetooth, ZigBee, USB, HTTP, CoAP, MQTT, etc.

2.1.1 ISO/IEEE 11073-20601

ISO/IEEE 11073-20601 optimized exchange protocol is located in the upper communication protocol layer and consists of application layer service and data exchange protocol. Application layer service provides a protocol for reliable data transfer and connection management. Data exchange protocol defines commands, PHD information and data format. ISO/IEEE 11073-20601 consists of three models; Domain Information model, Service model and Communication model [2].

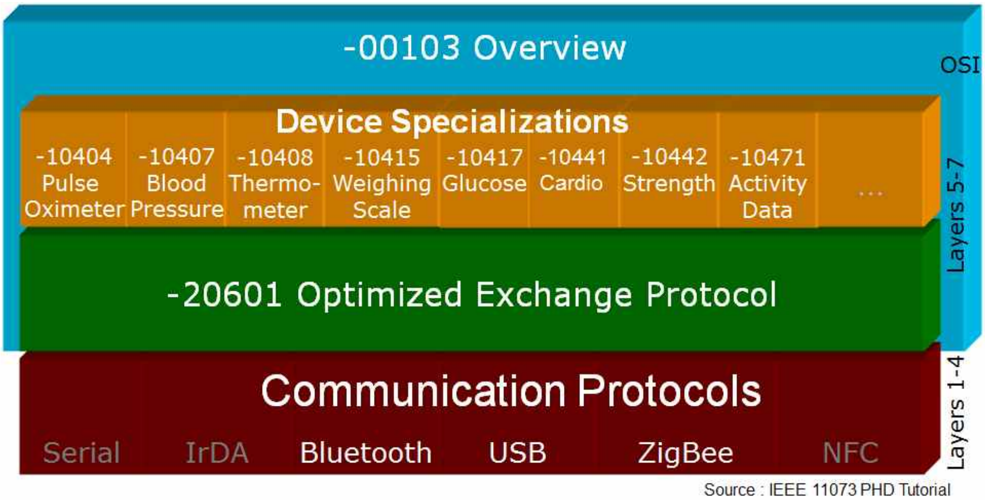

Figure 3 ISO/IEEE 11073 PHD Architecture

- **Domain Information Model**

DIM characterizes information from an agent as a set of objects. Each object has one or more attributes and the attributes describe measurement data that are communicated to a manager as well as elements that control behavior and report on the status of the agent [2].

- **Service Model**

The service model provides data access primitives that are sent between the agent and manager to exchange data from the DIM. These primitives include commands such as GET, SET, ACTION, and Event Report [2].

- **Communication Model**

The communication model supports the topology of one or more agents communicating over point-to-point connections to a single manager. For each point-to-point connection, the dynamic system behavior is defined by a connection state machine. The connection state machine defines the states and sub-states an agent and manager pair passes through, including states related to connection, association, and operation. The communication model also defines in detail the entry, exit, and error conditions for the respective states including various operating procedures for measured data transmission. The communication model also includes assumptions regarding the underlying communication layers' behavior [2].

DIM uses ASN.1 to define each attribute data type and exchange formats. Normal ASN.1 follows encoding rules such as BER(Basic Encoding Rules), CER(Canonical Encoding Rules), PER(Packed Encoding Rules), XER(XML Encoding Rules), etc. Figure 4 shows, MDS (Medical Device System) object of IEEE 11073-10407 blood pressure monitor using XER [11].

```xml
<?xml version="1.0" encoding="UTF-8"?>
<data-list>
  <entry>
    <compound>
      <name>MDS</name>
      <entries>
        <entry>
          <meta-data>
            <meta name="attribute-id">2438</meta>
          </meta-data>
          <compound>
            <name>System-Type</name>
            <entries>
              <entry>
                <simple>
        <name>code</name>
```

```xml
          <type>intu16</type>
          <value>0</value>
        </simple>
      </entry>
      <entry>
        <simple>
          <name>partition</name>
          <type>intu16</type>
          <value>0</value>
        </simple>
      </entry>
    </entries>
  </compound>
</entry>
<entry>
  <meta-data>
    <meta name="attribute-id">2344</meta>
  </meta-data>
  <compound>
    <name>System-Model</name>
    <entries>
      <entry>
        <simple>
          <name>manufacturer</name>
          <type>string</type>
          <value></value>
        </simple>
      </entry>
      <entry>
        <simple>
          <name>model-number</name>
          <type>string</type>
          <value></value>
        </simple>
      </entry>
    </entries>
  </compound>
</entry>
<entry>
  <meta-data>
    <meta name="attribute-id">2436</meta>
  </meta-data>
  <simple>
    <name>System-Id</name>
    <type>hex</type>
    <value>4C4E494147454E54</value>
```

```
          </simple>
        </entry>
        <entry>
          <meta-data>
            <meta name="attribute-id">2628</meta>
          </meta-data>
          <simple>
            <name>Dev-Configuration-Id</name>
            <type>intu16</type>
            <value>700</value>
          </simple>
        </entry>
        <entry>
          <meta-data>
            <meta name="attribute-id">2349</meta>
          </meta-data>
          <compound>
            <name>Production-Specification</name>
            <entries></entries>
          </compound>
        </entry>
      </entries>
    </compound>
  </entry>
</data-list>
```


Figure 4 IEEE 11073-10407 MDS Message

2.1.2 ISO/IEEE 11073-104xx

ISO/IEEE 11073-104xx device specializations standards are located in the upper ISO/IEEE 11073-20601 standard. This standard defines how to operate and present PHD. Table x shows a list of device specialization.

Table 1 Kind of 11073-104xx Standards

IEEE Std 11073-10404	Pulse Oximeter
IEEE Std 11073-10407	Blood Pressure Monitor
IEEE Std 11073-10408	Thermometer
IEEE Std 11073-10415	Weighing Scale
IEEE Std 11073-10417	Glucose Meter
IEEE Std 11073-10420	Body Composition Analyzer
IEEE Std 11073-10421	Peak Flow
IEEE Std 11073-10441	Cardiovascular Fitness and Activity Monitor
IEEE Std 11073-10442	Strength Fitness Equipment
IEEE Std 11073-10471	Independent Living Activity Hub
IEEE Std 11073-10472	Medication Monitor
IEEE Std 11073-10406	Basic ECG
IEEE Std 11073-10413	Respiration Rate Monitor
IEEE Std 11073-10418	INR(Blood Coagulation)
IEEE Std 11073-10419	Insulin Pump

3 Internet Protocols for IoT Healthcare Services

For providing IoT services, many global companies have developed IoT standard. Of those, oneM2M made by 8 organization for standardization (TTA, ETSI, TIA, ATIS, CCSA, etc.) and more 200 company, for developing IoT standard technical [12].

oneM2M's protocol working group uses application layer protocols in IoT environment like HTTP, MQTT, CoAP. These protocols can be adapted in a healthcare environment.

In IoT environment, some IoT devices have constrained characteristic like energy constraints, memory limitations, limit processing capability, unattended network operation, etc. Due to these constraints, IoT devices require light weight protocols like CoAP and MQTT.

3.1 MQTT

MQTT protocol has been in use since 1999 and was developed by IBM. The OASIS(Organization for the Advancement of Structured Information Standards) Technical Committee for the standardization of MQTT was established for organizations from around the world to collaborate and develop a standardized version of the MQTT protocol [13].

MQTT is Publish/Subscribe based light weight messaging protocol for use on top of the TCP/IP protocol. It is designed for connections to remote locations where a small code footprint is required or the network bandwidth is limited. The Publish/Subscribe messaging pattern requires a message broker[14].

Figure 5 shows a signaling flow diagram of MQTT. MQTT consists of publisher client, subscriber client and message broker. Publisher client can register a topic and publish messages on a topic. If subscriber client wants to subscribe to a specific topic, it requests for subscription to the message broker. After message broker receives a subscription request, then a session with subscriber client is established.

Thereafter, the message is published from publisher client to message broker and then forwarded to the subscriber client.

If the client does not need to send or receive messages, it has to send a keep-alive message to the broker so that a session is maintained. Without the broker receiving a keep-alive message, a session is disconnected between client and broker.

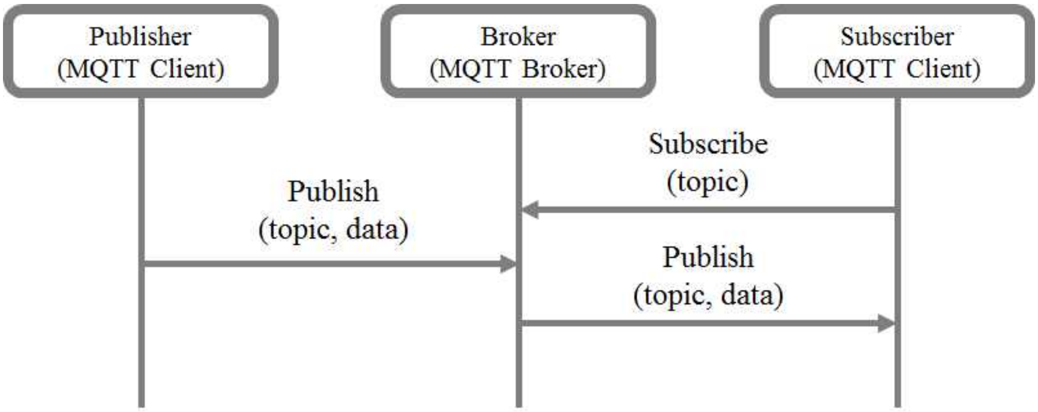

Figure 5 Signaling Flow of MQTT

To ensure the reliability of messaging, MQTT supports 3 levels of Quality of Services (QoS). Table 2 shows 3 different QoS levels [15].

Table 2 Quality of Service of MQTT

QoS Level	Delivery	Guarantee
0	At most once	Best Effort
1	At least once	Guaranteed
2	Exactly once	Guaranteed & No Duplicate

- Qos Level 0 : Sends message only once following the message distribution flow, and does not check whether the message has arrived to its destination. Therefore, in the case of sizable messages, it is possible that the message will be lost when any kind of loss comes in the way.

- Qos Level 1 : sends the message at least once, and checks the delivery status of the message by using the status check message, PUBACK. However, when PUBACK is lost, it is possible that the server will send the same message twice, since it has no confirmation of the message being delivered.

- Qos Level 2 : passes the message through exactly once utilizing the 4-way handshake. It is not possible to have a message loss in this level, but due to the complicated process of 4-way handshake, it is possible to have relatively longer end-to-end delays.

MQTT is concerned only with message transmission and it is the implementer's responsibility to provide appropriate security features. This is commonly achieved by using SSL/TLS [16].

As mentioned above, MQTT protocol has pros like overhead minimization, supports 3 levels of QoS and many to many communication, etc. However, MQTT has a disadvantage of session maintenance for message transfer. This is because the connected device never gets a rest time for it is always connected. The presence of a broker in the MQTT structure leads to a limit for providing various services.

3.2 CoAP

The CoAP(Constrained Application Protocol) is a specialized web transfer protocol for use with constrained nodes and constrained networks in the Internet of Things. CoAP became an international standard in 2014 designed by IETF CoRE(Constrined RESTful Environment) WG. CoAP is also used by various standard organizations like oneM2M, OMA, etc.

CoAP has been designed with the aim to be easily used in REST based architectures. Due to this, CoAP easily interoperates with HTTP through an intermediary proxy which performs cross-protocol conversion [17].

Figure 7 shows CoAP protocol stack. CoAP logically uses a two-layer approach, a CoAP messaging layer used to deal with UDP and the asynchronous nature of the interactions, and the request/response interactions using method and response codes [18].

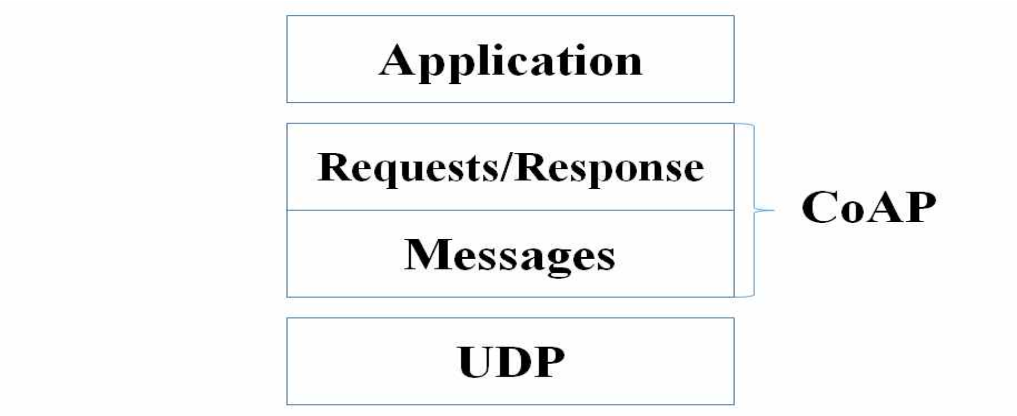

Figure 7 Abstract layering of CoAP

CoAP is composed of maximum 8 bytes default header and option header. CoAP message size is 10% smaller then HTTP because it uses binary encoding. Each CoAP message contains a Message ID that is used to detect duplicates and for optional reliability. It supports the RESTful architecture and expresses all of the resources in the URI. It uses the basic methods(GET, PUT, POST, DELETE) and Observe methods to define the act of its resources. This structure of CoAP has the advantage that it performs easily by interoperating with conventional HTTP.

CoAP provides CON(Confirmable) and NON(Non-Confirmable) message types for reliable message transfer. Figure 8 shows signaling flow diagram for each CoAP message type. A Confirmable message is retransmitted using a default timeout and exponential back-off between retransmissions, until the recipient sends an ACK message with the same Message ID from the corresponding endpoint. A message that does not require reliable transmission can be sent as a NON message and these are not acknowledged[18].

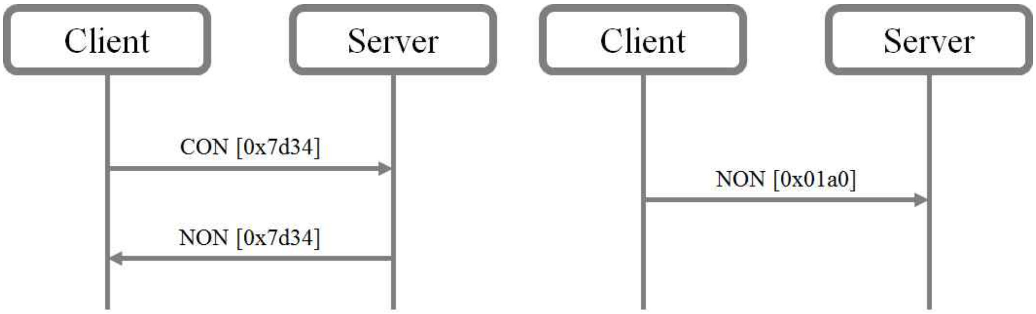

Figure 8 Signaling Flow of CoAP

CoAP has advantages of using DTLS to enhance its security, cross protocol proxy between CoAP and HTTP, discovery, etc. Due to CoAP advantages over other protocols, it is therefore the most suitable protocol to be used in IoT healthcare environment.

4 Integrating ISO/IEEE 11073-20601 and Internet Protocols

Recently, many researches are being carried out to combine internet protocols and ISO/IEEE 11073-20601 protocol in the IoT fraternity. Among many internet protocols, MQTT and CoAP are suitable in constrained IoT environment.

In [3], the ISO/IEEE 11073 communication model was adapted to the MQTT protocol, and thus enabling publish/subscribe model. Other works have been done to integrate CoAP protocol with ISO/IEEE 11073 PHD, such as [4], [5].

However, the works in [3] states out the advantages of supporting many to many communications, Quality of Service and small overhead, but has a disadvantage of limiting the rest time of devices because a connection for data transfer is always maintained. The works of [4] provides the limitation of only providing one-to-one communication for 11073 Event Report model. In [5], there is a limitation of constrained PHD rest time, because of the use of a CoAP server on the PHD, and therefore, PHD is always waiting to receive requests.

Ⅲ. Architecture of Pub/Sub Messaging System for IoT Healthcare Services Using CoAP

In this paper, we proposed CoAP based Pub/Sub messaging system for IoT healthcare services. This system provides a Publish/Subscribe structure as well as the existing Server/Client structure.

In this chapter, we describe the structure of CoAP based Pub/Sub messaging system, protocol stack and detail signaling flow diagram.

1. Integrating ISO/IEEE 11073-20601 and CoAP

In this paper, we proposed pub/sub messaging system for IoT healthcare services that merges ISO/IEEE 11073-20601 standard and CoAP standard. In the proposed system CoAP Client acts both as an 11073 agent and 11073 manager while the CoAP server acts as a message broker between the agent and the manager. JSON message format was used to represent DIM which acts as PHD's information. Table xxxxx represents a mapping of service model functions into CoAP methods. For example 11073 GET function is used by the manager to request for DIM that exists in the agent. We were able to map the GET function to the CoAP GET method in the proposed system.

Table 3 11073 Service Model Mapping to CoAP

11073 Service Model Function	CoAP Method
GET	GET
SET	PUT
ACTION	PUT
EVENT REPORT	Observe

2. System Architecture

Figure 9 shows, a CoAP Based Publish/Subscribe messaging system. CoAP Based Publish/Subscribe messaging system is composed of three parts: Publisher, Subscriber and message Broker.

Basically publisher, subscriber and message broker has the same operations as MQTT. CoAP Client acts as publisher and subscriber. Publisher registers a topic and publishes messages on a topic. Subscriber subscribes for a topic and receives published messages. Examples of CoAP Clients are healthcare data measuring device, treatment device and phone or web clients for diagnosis or management, etc.

CoAP Server acts as a message broker. The roll of CoAP server is to forward messages between clients based on a topic. More specifically, the broker receives topic requests from clients and registers them. When subscriber requests a topic, the broker receives the request and stores the subscriber's information in subscribers list.

As mentioned above, this system architecture has advantage of providing not only existing server/client model(1-to-1 communication) but also publish/subscribe model(many-to-many communication).

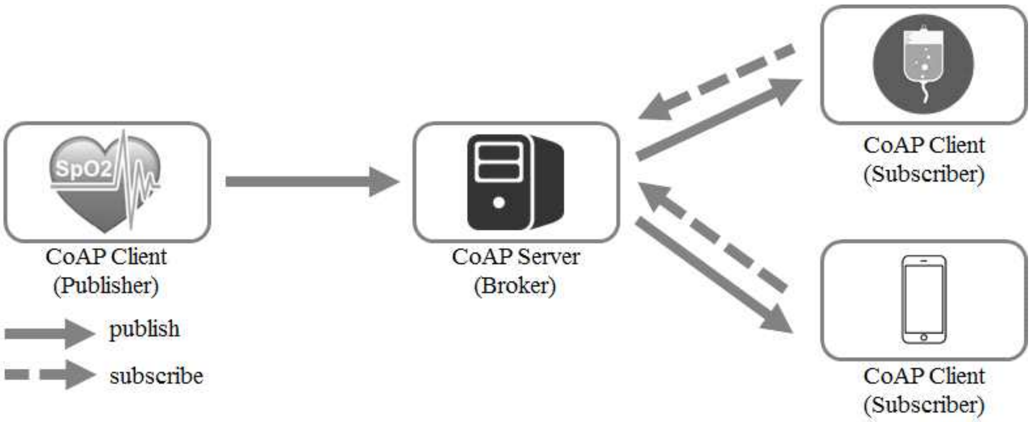

Figure 9 CoAP based Publish/Subscribe messaging system structure

3. Protocol stack for healthcare services in constrained IoT environment

Figure 5 shows protocol stack of healthcare services in constrained IoT environment. The ISO/IEEE 11073 DIM and CoAP are used in the application layer, UDP and DTLS are used in transport layer, and finally IPv4 or 6LoWPAN can used in IP Layer.

Specifically, ISO/IEEE 11073 DIM is used to represent healthcare data while CoAP is used for communication in a limited environment. DTLS and UDP are used for message encryption and it is transmitted from the upper layer. The IP layer may use 6LoWPAN (IPv6 over Low-power Wireless Personal Area Network) and the PHY / MAC Layer can use WIFI, 3G and Bluetooth, etc depending on the device type.

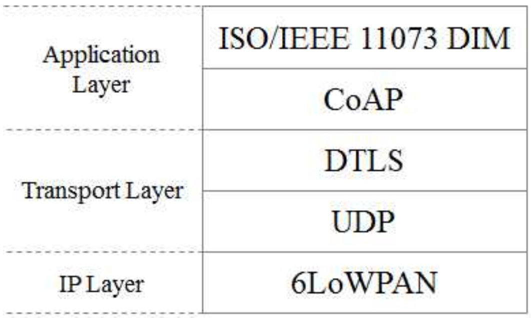

Figure 10 Protocol stack of proposed system

4. Signaling Flow of System

Figure 11 and 12 show a signaling flow diagram for CoAP based Publish/Subscribe messaging System. We assumed two service scenario.

- Scenario A : Monitoring

  Here the healthcare measurement device (heartbeat monitors, SPO2 monitor, etc) measures data from the patient. This data is forwarded to the message broker which is further forwarded to the treatment device or/and manager.

  Treatment device refers to a device that is capable of controlling what operations can be taken based on the received healthcare information from the measurement device. For example, infusion regulator, patient temperature controller, etc are some of the treatment devices.

  Manager means a device capable of continuously monitoring the received healthcare information from the measurement device. For example, doctors and nurses smartphone and patient information recording devices are managers.

  The detailed signal flow description of the first scenario is as follows.

Topic registration:

Healthcare measurement device uses CoAP POST method for topic registration. Registered topic resource is represented by a URI. For example coap://brokeraddress.com/Device_A/Topic_Container/Measurment is a topic resource about a healthcare measured information of a device.

Topic Discovery:

Subscriber devices need to know the topic resource name for subscription. To do this, subscriber devices can request using CoAP GET method to well-known resource of message broker for finding topic resources. Due to the received discovery request by the broker from the subscriber clients, the broker then sends topic resource list as a response message to the subscriber client.

Request Subscription:

Devices can subscribe for a topic resource using CoAP Observe method. On the broker receiving a subscription request from a device, it saves device end-point information that will enable the broker to send the published data to that device. In the above process, the initial set of procedures for the Publish / Subscribe system comes to an end at this point.

Publish Message:

Here the broker receives new data on a specific topic resource from the healthcare measurement device and saves it in the database. For this case the healthcare measuring device uses CoAP PUT method for publishing the data.

Since the end-point information for the subscribers have already been saved by the message broker, on the message broker receiving the published data, it forwards published data to the subscribers using their end-point information.

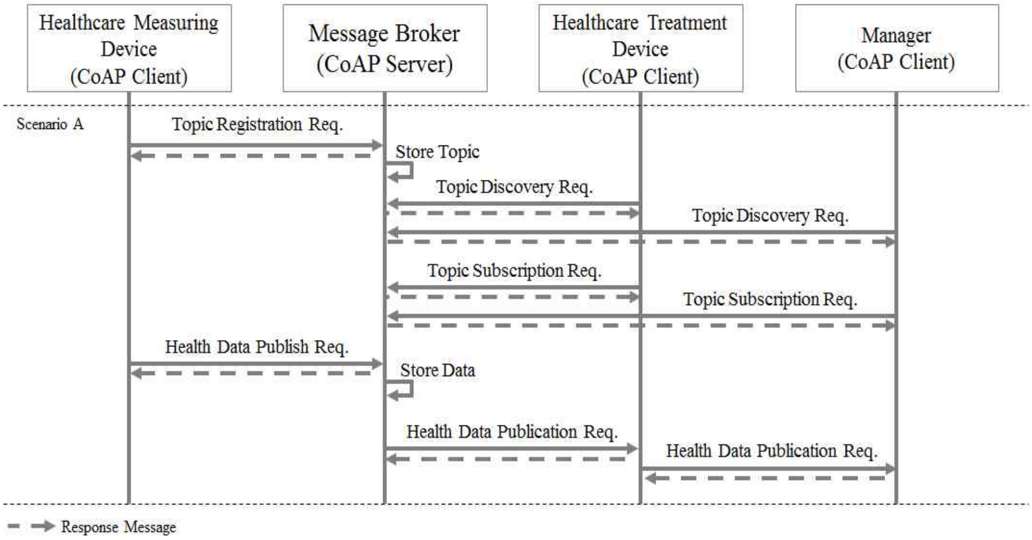

Figure 11 Signaling flow diagram

- Scenario B : Controlling

  In scenario B, the medical staff remotely controls the healthcare treatment device.

  The manager can register topic resource for control command by following the same steps as in Scenario A, topic registration. The treatment device also subscribes for this same control command resource by also following the same steps as in scenario A, request subscription.

  Following these control steps, the manager comfortably transmits a remote control command to the health treatment device without knowing the location of the health treatment device.

Table 3 shows the methods of CoAP that are used in the proposed system. CoAP GET method is used for the topic resource discovery. CoAP POST method is used for topic resource registration. CoAP PUT method is used for publishing measured health data or control message to a message broker. CoAP DELETE method is used for topic resource deletion when not used. Finally, CoAP Observe method is used for topic resource subscription.

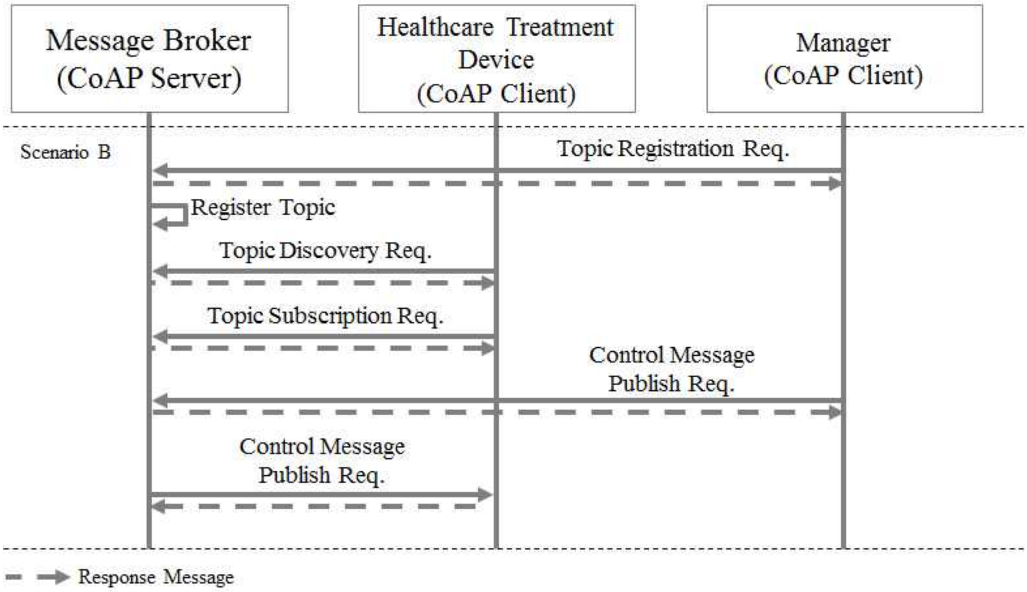

Figure 12 Signaling flow diagram

Table 4 Used CoAP method list

CoAP Function	Explanation
GET	Resource Discovery, Request Resource Data
POST	Topic resource Registration
PUT	Publish Data
DELETE	Topic resource Deletion
Observe	Request Subscription

Ⅳ. Implementation of Pub/Sub Messaging System for IoT Healthcare Services Using CoAP

In this section, we implemented a prototype of a CoAP based Publish/Subscribe messaging system for IoT healthcare services. We hereby present the implementation environment and result of the prototype for the proposed system. Thereafter, performance of the system has been evaluated by comparing with publish/subscribe system based on MQTT.

1. Implementation Environment

We implemented a realistic prototype for our proposed messaging system for IoT healthcare as shown in Figure 13. We evaluated the performance and verified the feasibility of proposed system. In the experiment, two raspberry pi, a spo2 sensor, one android smart phone, one desktop computer and one AP were used. The raspberry pi acted as the healthcare measurement device and healthcare treatment device. The desktop computer acted as the CoAP message broker. The android smart phone acted as healthcare manager. Finally, all the different components were connected in one AP.

The component specifications are as follows:

Table 5 Specifications for the components used to implementation

Component	Environment
Raspberry Pi (Healthcare Measurment Device, Healthcare, Healthcare Treatment Device)	Raspberry Pi 2, spo2 sensor(HBE-spo2), Californium CoAP lib
Smartphone (Healthcare Manager)	Nexus 5[LGD821 model] (OS version 5.1)
Desktop Computer (CoAP Message Broker)	Intel i7, 10 GB ram, 64bit Windows 10
Access Router	54Mbps Wireless Router WGR614 v8
Development language	Java, Python

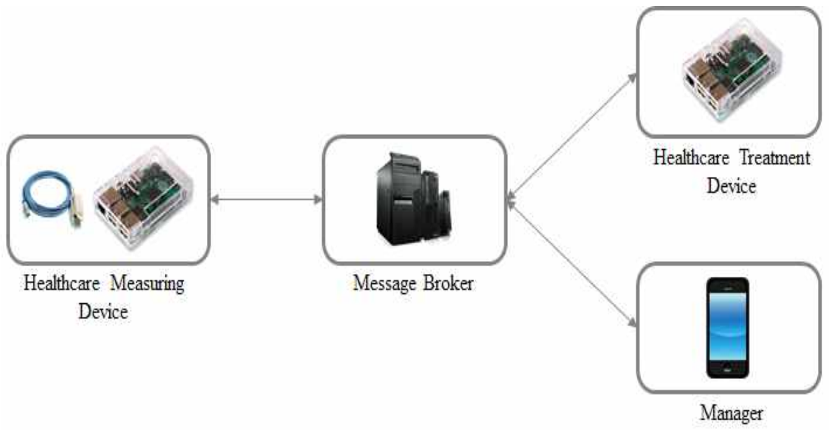

Figure 13 Prototype for CoAP based publish/subscribe messaging system for IoT Healthcare services

2. Implementation Result


```
{
    METRICS:
    [ (length 2)
        {
            NUMERIC-SpO2:
            [ (length 1)
                {
                    MDC_ATTR_METRIC_SPEC_SMALL: 53312
                    MDC_ATTR_ATTRIBUTE_VAL_MAP: 2636
                    MDC_ATTR_ID_HANDLE: 1
                    MDC_ATTR_NU_VAL_OBS_BASIC: 90
                    MDC_ATTR_UNIT_CODE: 544
                    MDC_ATTR_ID_TYPE: 19384
                }
            ]
        }
        {
            NUMERIC-Pulserate:
            [ (length 1)
                {
                    MDC_ATTR_METRIC_SPEC_SMALL: 53312
                    MDC_ATTR_ATTRIBUTE_VAL_MAP: 2636
                    MDC_ATTR_ID_HANDLE: 10
                    MDC_ATTR_NU_VAL_OBS_BASIC: 79
                    MDC_ATTR_UNIT_CODE: 2720
                    MDC_ATTR_ID_TYPE: 18458
                }
            ]
        }
    ]
}
```


Figure 14 ISO/IEEE 11073 DIM JSON message

Figure 14 shows JSON format of ISO/IEEE 11073 DIM used for transferring the measured medical information from a PHD. ISO/IEEE 11073 DIM was created by referencing to ISO/IEEE 11073-10404 document. ISO/IEEE 11073-10404 document defines pulse oximeter device specialization. Pulse rate and oxygen saturation as were measured as shown in the DIM above.

Figure 15 illustrate CoAP message broker analysis using CoAP Copper plugin for Firefox extension program. Figure 15 contains message broker address at the top and resource list of the message broker at the left side. In the resource list, we can see MeasurementData topic resource and ControlData topic resource. At the center of figure 15, we can see response information from message broker.

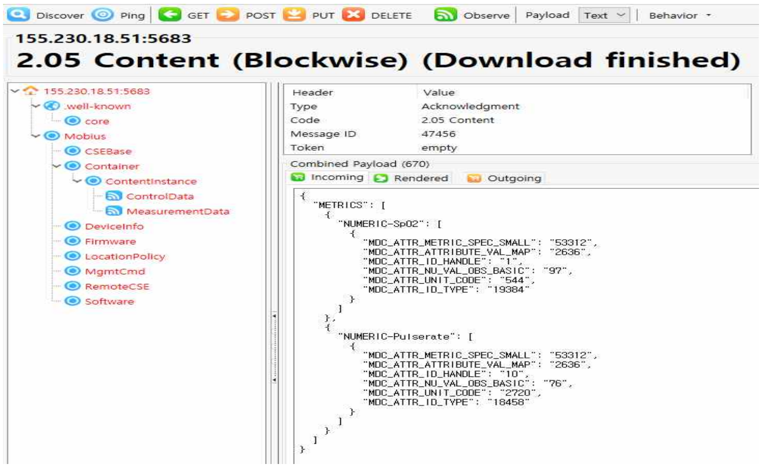

Figure 15 CoAP message broker analysis

Figure 16 shows, a screen shot of publisher client. If there is no registered topic resource on the message broker, publisher client needs to request for topic resource registration. Thereafter, publisher client publishes ISO/IEEE 11073 DIM message when healthcare data is measured from pulse oximeter sensor. This implementation result of the publisher client is shown in Figure 16.

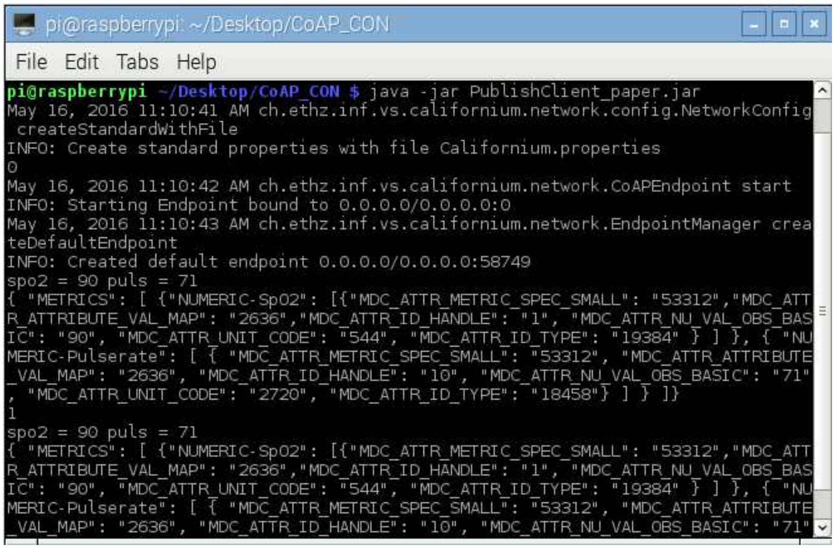

Figure 16 Screen shot of publisher client

Figure 17 and 18 shows subscriber clients screen shots. Each subscriber client is implemented on raspberry pi and smartphone.

We can therefore see the potential of the proposed messaging system through the implemented prototype.

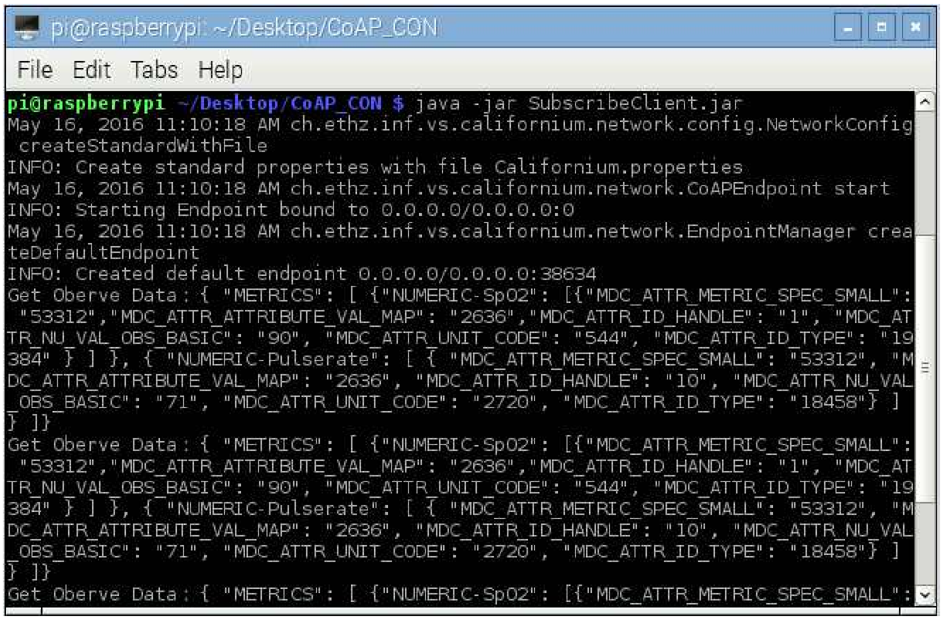

Figure 17 Screen shot of subscriber client

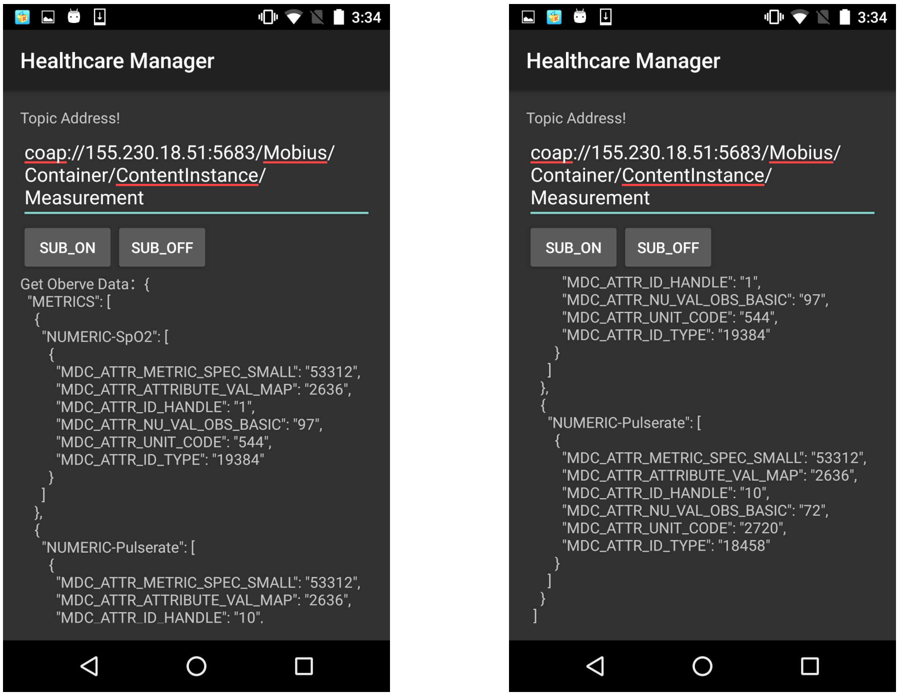

Figure 18 Screen shot of subscriber clients

V. Performance Evaluation

For the evaluation of the proposed messaging system, we compare and analyze CoAP based publish/subscribe messaging system with MQTT based publish/subscribe messaging system. To do this, we used Linux tc command to simulate packet loss rate as shown in Figure 19.

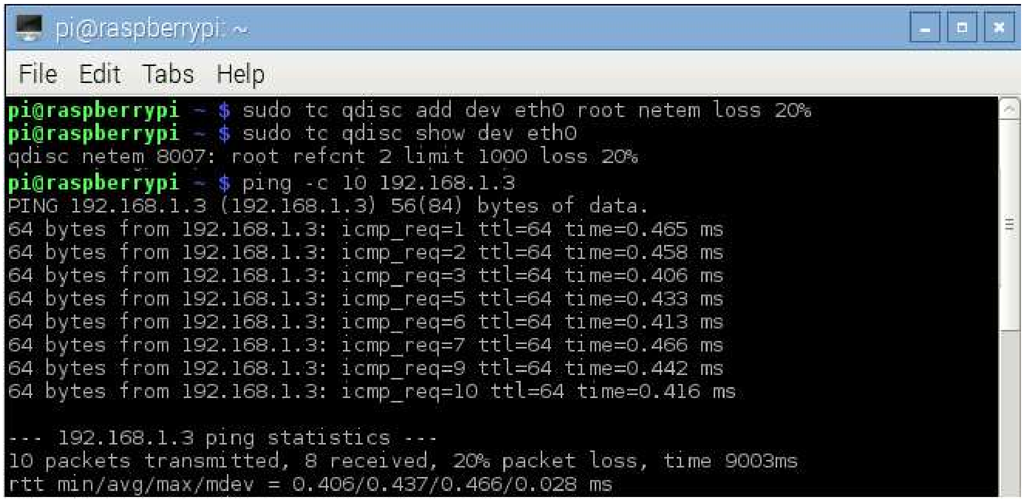

Figure 19 Setting packet loss rate with tc command

This performance test was executed to validate the key features of the proposed messaging system, such as light weight protocol, reliability provision, etc, In this simulation environment, the loss rate of 5, 10, 15, 20, 25, 30% were added. To provide reliability, CoAP protocol supports CON and NON option While MQTT protocol supports QoS 0, 1, and 2.

Fig. 20 shows, it illustrates average RTT (Round Trip Time) in one complete transaction. We used only the scenario with reliability (CoAP CON vs MQTT QoS 1 and 2). The results shows that the CoAP RTT is more shorter than MQTT. This result is not in any way based on the different performance effects of UDP and TCP.

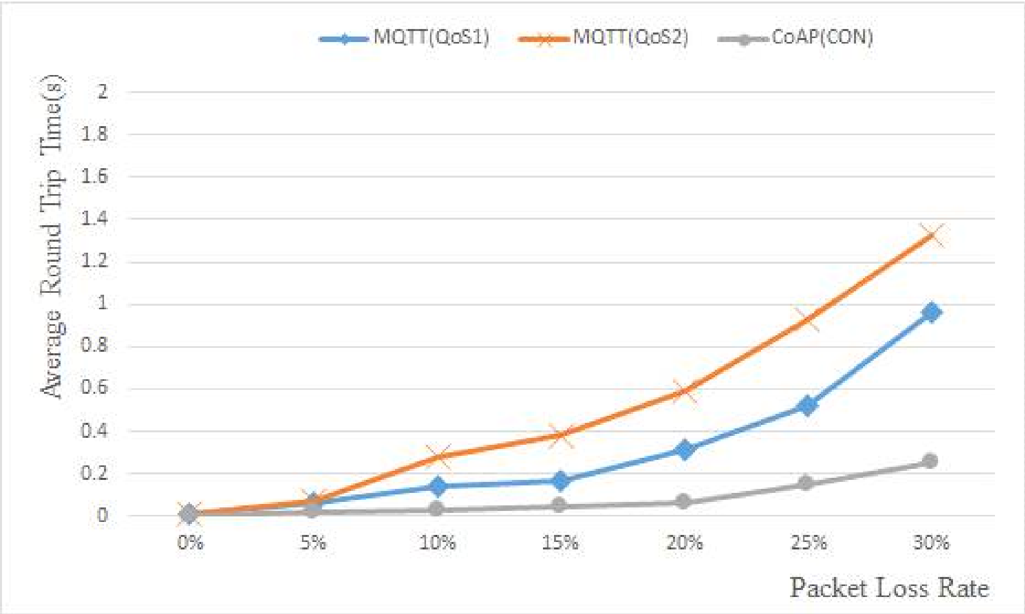

Figure 20 Average RTT of each packet loss rate

Fig. 21 shows the number of packets used when healthcare measurement device sends the measured biomedical message to the message broker and the message broker transfers biomedical message to treatment device and manager. For fair comparison, the exact same biomedical message was used for transfer, and each protocol's broker and clients were installed on the same desktop computer and each Raspberry Pi. We published the information once after every 10 seconds for ten times.

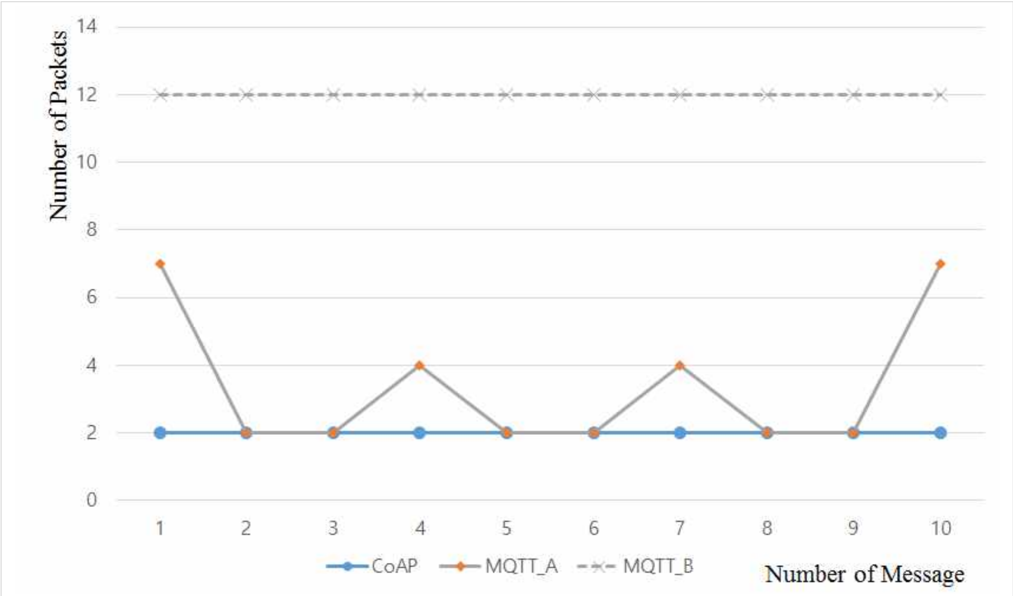

Figure 21 Number of packets for transfer

The results shows that CoAP protocol based system only used 2 packets for each transfer. If we transfer by NON option of CoAP, only 1 packet can be used for the transfer, but the reliability falls at the same time. MQTT_A transfers the message 10 times on single connection. To do this, first message requires 7 packets for 3-way handshaking and MQTT connection packet, and 2 additional ping packets are used periodically to hold persistent connection. Also, to make the last termination, 7 packets are used for MQTT connection termination packet and 4-way handshaking. MQTT_B client establishes and terminates the connection each time when the message is transferred, using 12 packets for each transfer.

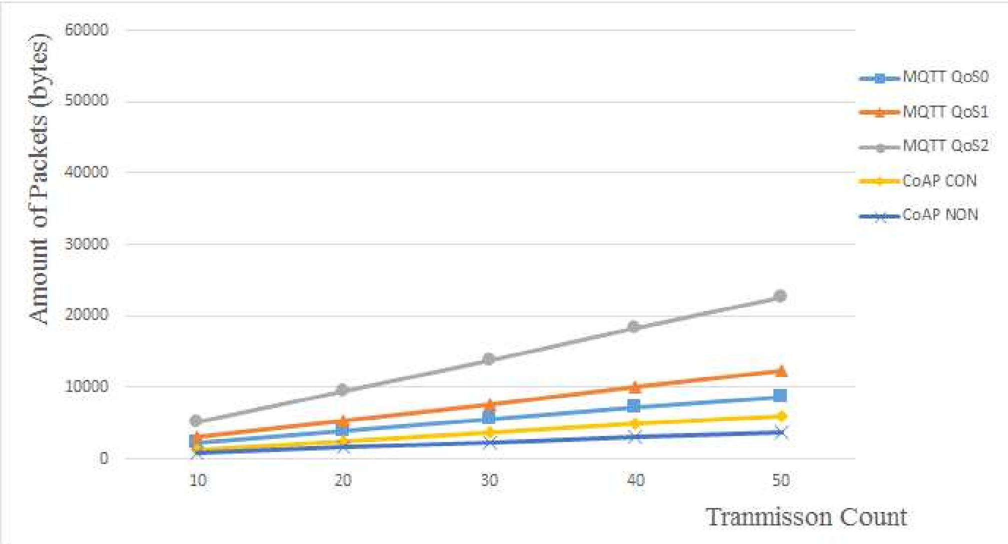

Figure 22 Amount of packets in subscriber

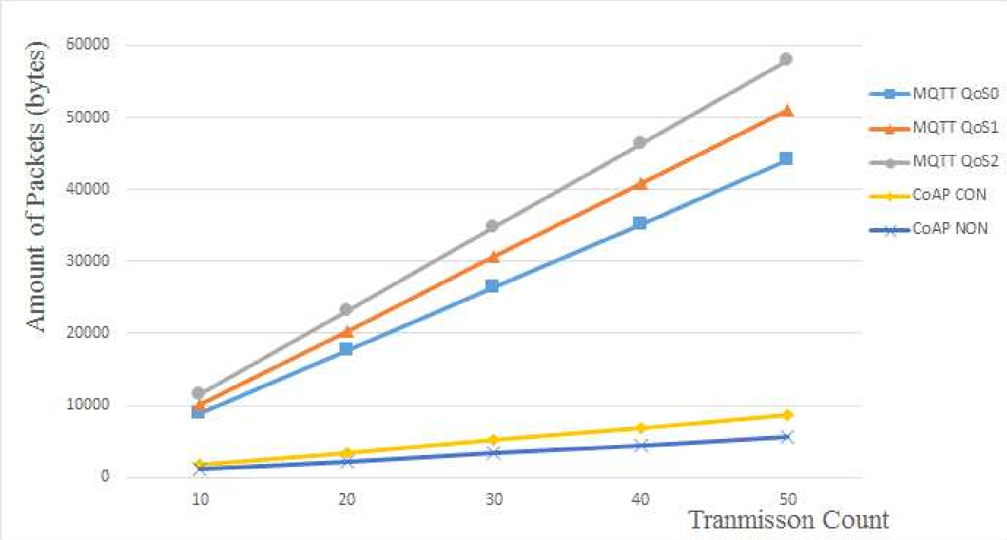

Figure 23 Amount of packets in publisher

In Fig 22 and 23, it illustrates how many packets were transmitted in each transaction, based on MQTT solution, it needs more packets than CoAP solution. The reason is that compared with CoAP a UDP based protocol MQTT is a protocol which is based on TCP and this means that every MQTT connection needs to start with 3-way handshaking to start TCP connection and finish with 4-way handshaking. Looking at the results in Figure 22 and 23, as the Transmission count increases, amount of packets increases steadily when MQTT is used than CoAP. This means that MQTT will use more battery of device than CoAP.

Ⅵ. Conclusion

In this paper, we have presented a Publish/Subscribe messaging System for IoT healthcare Services using CoAP. We have designed an architecture, and a detailed signaling procedures for a CoAP based Publish/Subscribe messaging system. Finally, prototype of proposed system was implemented, and a comparative performance evaluation between CoAP based system and MQTT based system in terms of round trip time and amount of transaction packets. The results shows that the proposed CoAP based messaging system is superior to previous messaging systems. This study was focused mostly on healthcare. In the future, we hope to extend to various industries such as smart factory, smart farm, smart city, etc.

.

Reference

[1] U. Mulligan and R. Brennan, “oneM2M: The Platform Standard for the Internet of Things”, Sand Hill, Jan, 2015.

[2] IEEE. ISO/IEEE 11073-20601: IEEE Health informatics – Personal health device communication - Part 20601:Optimized exchange protocol Standards.

[3] Gomes, Yuri F., et al. "Integrating MQTT and ISO/IEEE 11073 for health information sharing in the Internet of Things." Consumer Electronics (ICCE), 2015 IEEE International Conference on. IEEE, 2015.

[4] Santos, Danilo FS, et al. "Integrating IEEE 11073 and constrained application protocol for personal health devices." Proceedings of the 29th Annual ACM Symposium on Applied Computing. ACM, 2014.

[5] S. Ge, et al. "Design and implementation of interoperable IoT healthcare system based on international standards." 2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC). IEEE, 2016.

[6] The Continua Health Alliance http://www.continuaalliance.org

[7] Wartena, Frank, Johan Muskens, and Lars Schmitt. "Continua: The impact of a personal telehealth ecosystem." eHealth, Telemedicine, and Social Medicine, 2009. eTELEMED'09. International Conference on. IEEE, 2009.

[8] Health Level Seven International. [Online]. Available: http://www.hl7.org/v3ballot/html/infrastructure/rim/rim.htm

[9] Mustafa Yuksel and AsumanDogac, “Interoperability of Medical Device Information and the Clinical Applications: An HL7 RMIM basedon the ISO/IEEE 11073 DIM”, IEEE TRANSACTIONS ON INFORMATION TECHNOLOGY IN BIOMEDICINE, VOL. 15, NO. 4, JULY 2011, PP 557

[10] Park, Chan-Yong, Joon-Ho Lim, and Soojun Park. "ISO/IEEE 11073 PHD standardization of legacy healthcare devices for home healthcare services." Consumer Electronics (ICCE), 2011 IEEE International Conference on. IEEE, 2011.

[11] Antidote-Signove. [Online]. Available: http://oss.signove.com/index.php/Antidote:_IEEE_11073-20601_stack

[12] Swetina, Jorg, et al. "Toward a standardized common M2M service layer platform: Introduction to oneM2M." Wireless Communications, IEEE 21.3 (2014): 20-26.

[13] Chen, Whei-Jen, et al. Responsive Mobile User Experience Using MQTT and IBM MessageSight. IBM Redbooks, 2014.

[14] MQTT in Wikipedia http://en.wikipedia.org/wiki/MQTT

[15] Lee, Shinho, et al. "Correlation analysis of MQTT loss and delay according to QoS level." Information Networking (ICOIN), 2013 International Conference on. IEEE, 2013.

[16] Banks, Andrew, and Rahul Gupta. "MQTT Version 3.1. 1." OASIS Standard (2014).

[17] Castellani, A., et al. "Guidelines for HTTP-CoAP Mapping Implementations." draft-ietf-core-http-mapping-05 (2014).

[18] Shelby, Z., K. Hartke, and C. Bormann. "The Constrained Application Protocol (CoAP)(RFC 7252)." (2014).

사물인터넷 헬스케어 서비스를 위한 CoAP 기반 메시징 시스템 설계 및 구현

김 현 수

경북대학교 대학원 전자공학부 정보통신공학전공
(지도교수 박종태)

(초 록)

최근 사물인터넷에 대한 기대와 관심이 증가하고 있다. 이러한 사물인터넷 기술은 스마트 카, 스마트 홈, 헬스케어 등 다양한 분야에 적용이 가능하다. 특히 100세 시대의 도래와 삶의 질에 대한 관심 증가로 질병대처의 패러다임이 치료에서 진단, 예방으로 전화도미에 따라 IoT 헬스케어 서비스가 점차 중요해지고 있다. IoT 환경의 다비이스들은 전력이 제한되어 있고, CPU처리 능력이나 전력 용량 등이 제한적이다. 이러한 제한적 특성 때문에 많은 사물인터넷 환경에서 데이터 전송을 위해 MQTT, CoAP 등과 같은 통신 프로토콜을 사용하고 있다. MQTT는 경량의 Publish/Subscribe 메시징 프로토콜로써, 다-대-다 통신이 가능하다는 장점이 있지만 전송을 위해 항시 연결을 유지하는 구조는 디바이스의 휴식 시간을 제한한다는 단점 때문에 제한적 IoT 환경에서 서비스를 제공하기에는 제한적이다. 이러한 문제점을 해결하기 위해, 본 논문에서는 IoT 헬스케어 서비스를 위한 CoAP 기반 메시징 시스템을 설계 하였다. 제안된 시스템의 실현 가능성을 검증하기 위해 ISO/IEEE 11073 DIM 메시지를 CoAP 기반 메시징 시스템을 사용한 프로토타입을 구현하였고, 성능 분석을 위해 기존 MQTT 기반 메시징 시스템과 제안한 CoAP 기반 메시징 시스템의 성능 비교를 하였다. 성능분석의 결과는 제안된 시스템의 성능을 검증하였다. 제안한 시스템은 헬스케어 뿐만 아니라 다양한 산업에 활용 될 수 있을 것이다.