Design and Implementation of an Interoperable Messaging System for IoT Healthcare Services

Brian Oryema, Hyun-Su Kim, Wei Li and Jong Tae Park
School of Electronics Engineering
Kyungpook National University
Daegu, South Korea
boryema@knu.ac.kr, hs_kim@ee.knu.ac.kr, liwei4899@naver.com and jtpark@ee.knu.ac.kr

***Abstract*— The spry maturation of Internet of Things (IoT) has paved way to the rapid development of numerous sectors and these have been envisioned in; connected transport, smart cities, connected homes, connected healthcare, etc. IoT is a technology that connects "things" that are embedded with sensors, actuators and network connectivity to collect and exchange the data to the internet. The ability of IoT that offers distinct technologies for a small constrained device to collect and deliver messages across sophisticated networks leaves room for more exploration. Zeroing down to the healthcare sector, a couple of Personal Health Devices (PHDs) have been developed to collect and share information across the internet. One Machine to Machine (oneM2M), ISO/IEEE 11073 PHD are some of the healthcare standards that have been developed have been developed to deal with the issue of interoperability in the IoT. In this paper we design and implement an interoperable messaging system that is based on international standards for IoT healthcare services. Standard IoT protocols; Message Queuing Telemetry Transport (MQTT) and Constrained Application Protocol (CoAP) are some of the protocols that were designed to be used in an IoT environment. We designed and implemented a message system using CoAP following international standards for IoT healthcare services. This is due to better performance that CoAP offers in a constrained environment over other protocols. The paper further analyses and evaluates a comparative performance of number of packets transmitted in a transaction and packet loss rate number during transmission between the designed system and existing messaging system that uses MQTT.**

Keywords—IoT; PHD; oneM2M; ISO/IEEE 11073 PHD; MQTT; CoAP;

I. Introduction

The advancement of the Internet of Things has highly contributed for its implementation in various industries such as healthcare, smart city, smart factory, smart farm, etc. There has been a high demand for better ways of monitoring personal health for some time now, and due to this, it has geared up the development of IoT healthcare services globally in the healthcare sector.

A couple of studies are being conducted unceasingly on how best IoT and healthcare technologies can be merged to provide healthcare services. One of the studies that has been performed on was the convergence of oneM2M and ISO/IEEE 11073 PHD standards. The use of international standards can help to solve problems on the existing vertical model 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 standard 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 widely used healthcare standards that does define how to send, monitor and control biometric information using a healthcare device [2]. ISO/IEEE 11073 PHD standard is an independent transport protocol, thus its implementation is shared between different devices and wireless or wired transport technologies implying that it can be transported either using Bluetooth, ZigBee, USB or any other means. ISO/IEEE 11073 PHD does not consider internet protocols in the under layer communication protocol. Therefore, there is need for the integration of ISO/IEEE 11073 PHD to oneM2M protocols in order for easy IoT healthcare services provision. Never the less, various research studies have been done to try and solve the issues stated above. In [3], the ISO/IEEE 11073 communication model was integrated into MQTT protocol for health information sharing in the internet of things and thus enabling a publish/subscribe model structure of message operation. Other works have been performed to integrate CoAP protocol into ISO/IEEE 11073 PHD health standard in [4], [5].

The work in [3] states out the advantages of MQTT being able to support many to many communications, Quality of Service(QoS) and having a small overhead, but having a disadvantage of limiting the rest time of devices when transmitting data. This is because for data transfer, a seamless connection has to be maintained. The work in [4] has also the limitation of only providing one-to-one communication for 11073 Event Report model. In [5], a CoAP server is embedded on to the PHD and this implies that the PHD is always waiting for requests, thus not getting rest time. This is a serious limitation to a constrained device with constrained resources in the IoT. In order to solve the drawbacks that other protocols offer in IoT healthcare, in this paper, a design and implementation of an interoperable messaging system for IoT healthcare services has been developed and analyzed. The proposed system is designed to combine the advantages that CoAP and MQTT offer in constrained environments; small

978-1-5090-6196-9/17/$31.00

message size, QoS, etc., and the support of many to many communication in IEEE 11073 Event Report respectively.

II. Related Work

A. Healthcare Standards

The interoperability of PHDs is very important for provision of healthcare services. Therefore, a couple of international standards organizations have developed standards; 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 to enable development of healthcare services.

ISO/IEEE 11073 PHD is a standard that provides the interoperability between a personal health device and the health manager. This consists of the ISO/IEEE 11073-20601, and IEEE 11073-104xx. ISO/IEEE 11073 PHD defines the OSI protocol stack from layer 5 to 7 and is an independent transport protocol. This implies that any transport protocol; Bluetooth, ZigBee, USB, HTTP, CoAP, MQTT, etc. can be used for the transportation of ISO/IEEE 11073 PHD. ISO/IEEE 11073-20601 optimized exchange protocol is located in the upper communication protocol layer and consists of the 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 and these are the PHD information and the data format. ISO/IEEE 11073-20601 consists of three models; Domain Information model, Service model and Communication model.

The Domain Information Model (DIM) characterizes information from an agent as a set of objects and each object has one or more attributes. 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. 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. 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 goes through, including states related to the 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 to the underlying communication layers' behavior [2].

B. Internet Protocols in oneM2M

Many global companies have developed IoT standards to enable easy provision of IoT services. oneM2M is the leading standardization body for M2M and IoT.oneM2M comprise of 8 organizations for standardization and 200 companies, for developing technical IoT standard.oneM2M's protocol working group uses application layer protocols in the IoT environment; HTTP, MQTT, and CoAP. oneM2M's architecture and standards have been developed to be used in most of the sectors ranging from industrial automation, home automation to eHealth and telemedicine. Therefore without any doubt, oneM2M's protocols can be easily adapted in a healthcare environment setting. In an IoT environment, most of the IoT devices are constrained in nature in terms of limitation in memory, processing capability, network connectivity, etc. Therefore light weight protocols; CoAP or MQTT should be used for in order to counteract limitations of being constrained in an IoT environment.

MQTT is a Publish/Subscribe light weight messaging protocol for use on top of the TCP/IP protocol that was designed for connections to remote locations where a small code footprint is required or network bandwidth is limited. For any Publish/Subscribe messaging pattern there is a need for a message broker as one of the components [6]. Fig. 1 shows MQTT's signal flow Diagram and consists of a publisher client, a subscriber client and a message broker. MQTT is based on a publisher/subscriber model where a client has to connect to the broker and exchange messages. For any subscriber client that is interested for a topic, it has to subscribe for the messages on the broker to listen to the topics. If the client is not interested in sending or receiving 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 the client and the broker.

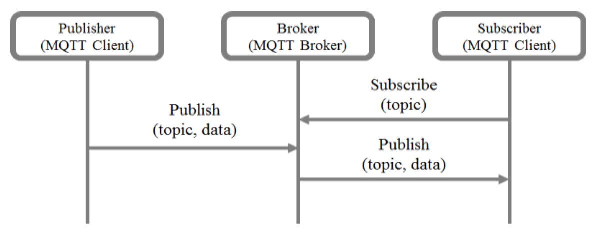


Fig 1. Signal Flow of MQTT

Table I shows three different QoS levels that MQTT supports in order to ensure message reliability.

In QoS Level 0, a message is sent only once following the message distribution flow, and does not check whether the message has arrived to its destination. Therefore, for sizeable messages, it is possible that the message will be. 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 exactly once utilizing the 4-way handshake. The possibility of experiencing a message loss in this level is almost zero. The complicated process of the use of 4-way handshake in QoS Level 2 leads to a relatively longer end-to-end delay. As mentioned earlier on, MQTT protocol has pros like overhead minimization, support of 3 levels of QoS and many to many communication, etc. However, MQTT has a disadvantage of maintaining a session for message transfer. This is because the connected device never gets a rest time because it always has to keep a connection. The presence of a broker in the MQTT structure leads to a limitation for various services provision.

CoAP (Constrained Application Protocol) is a specialized web transfer protocol for use with constrained nodes and constrained networks in the Internet of Things environment.

TABLE I. QUALITY OF SERVICE FOR MQTT

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

CoAP was designed with the aim to be used in REST based architectures. Due to this, CoAP easily interoperates with HTTP through an intermediary proxy which performs cross-protocol conversion [9]. Fig. 2 shows a CoAP protocol stack. CoAP logically uses a two-layer approach; a CoAP messaging layer that is based on UDP and the asynchronous nature of the interactions, and the request/response interactions using method and response codes [10].

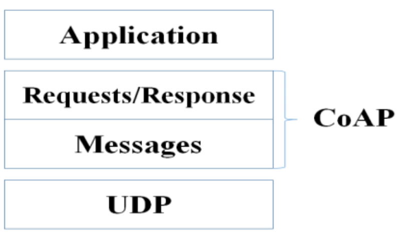


Fig 2. Abstract Layering of CoAP

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

CoAP provides CON (Confirmable) and NON (Non-Confirmable) message types for a reliable message transfer and Fig. 3 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 [10].

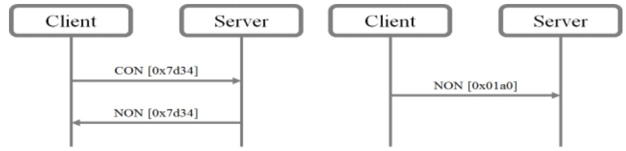


Fig. 3. Signal flow of CoAP

CoAP has advantages of using DTLS to enhance its security, cross protocol proxy between CoAP and HTTP, discovery, etc. Therefore, because of the advantages that CoAP has over other protocols, it has become the most widely used suitable protocol in the IoT healthcare environment.

III. DESIGN OF THE INTEROPERABLE MESSAGING SYSTEM BASED ON INTERNATIONAL STANDARDS FOR IOT HEALTHCARE SERVICES

A. System Architecture

Fig. 4 illustrates an architecture of the messaging system for IoT Healthcare services that has been proposed and it consists of three components; publisher, subscriber and message broker. All these components are integrated to perform different responsibilities for a common goal of offering a healthcare service in the IoT. CoAP Clients either act as a publisher or subscriber in the architecture. In this architecture all the clients are based on CoAP protocol because of the advantages that CoAP offers in the IoT fraternity.

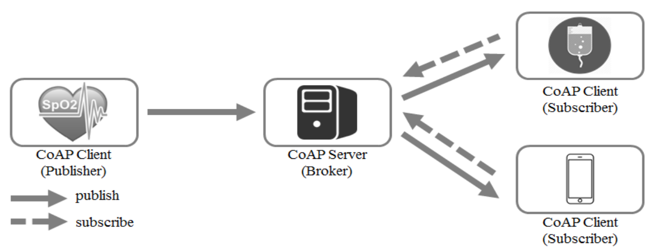


Fig 4. System Structure of the messaging system

A publisher is used to registers a topic and publish messages on a topic while a subscriber has a role of subscribing for topics on the broker and receive the published messages whereas the message broker forwards messages between clients based on a specific topic. CoAP server in this architecture acts as a message broker. The broker specifically receives topic requests from clients and registers them. When subscriber requests for a topic, the broker receives the request and stores the subscriber's information in subscribers list. Examples of CoAP Clients are healthcare data measuring devices, treatment devices and phone or web clients for diagnosis or management, etc. Basically the publisher, subscriber and message broker in this architecture all operate the same way they would operate in MQTT.

The strength of this system architecture is that it does not only support the existing one to one server/client model but further also supports many-to-many communication publisher/subscriber model.

B. *Protocol stack for healthcare services in constrained IoT environment*

The protocol stack used in the proposed messaging system architecture for the IoT healthcare services is shown in Fig. 5. The ISO/IEEE 11073 DIM and CoAP are used in the application layer, UDP and DTLS used in transport layer, and finally IPv4 or 6LoWPAN can be used in the IP Layer. ISO/IEEE 11073 DIM is used to represent healthcare data while CoAP is used for communication in a constrained environment. DTLS is used for message encryption and is transmitted from the upper layer for security purpose. The IP layer may use 6LoWPAN (IPv6 over Low-power Wireless Personal Area Network) and the PHY / MAC Layer may use WIFI, 3G and Bluetooth, etc. depending on the device type.

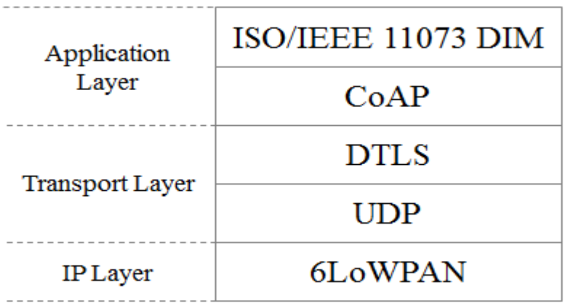


Fig 5. Protocol stack of the proposed system

C. *Signal Flow Diagram*

The signal flow of the proposed system was divided in to two scenarios; scenario A; Monitoring and scenario B; Controlling, and they are shown in Fig. 6 and 7 respectively.

1) Scenario A:

Monitoring: Here the healthcare measurement device (heartbeat monitors, SPO2 monitor, etc.) measures data from the patient. This data is then forwarded to the message broker which is further forwarded to the treatment device or/and manager. A treatment device is a device that is capable of controlling a specific condition that is based on the received healthcare information from the measurement device. For example, infusion regulator, patient temperature controller, etc. are some of the treatment devices. A manager is a device that is 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 scenario A; Monitoring is shown in Fig. 6.

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

Topic Discovery: For subscriber devices to be able to subscribe for a resource, then they should be knowing the resource name. To do this, the subscriber devices can request for finding a topic resource using CoAP GET method to a well-known resource of message a broker. 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 the device end-point information and this is what the broker uses 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 is already known by the message broker, on the message broker receiving the published data, it forwards the data to the subscribers using their end-point information.

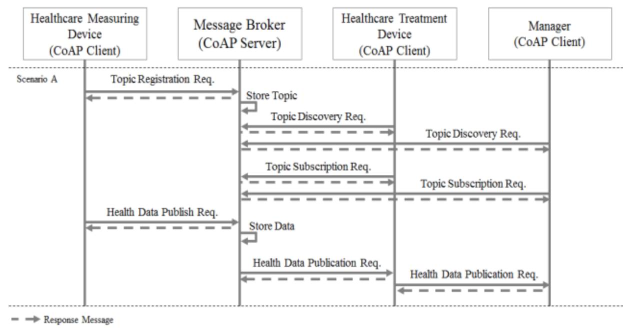


Fig 6. Scenario A: Monitoring signal flow diagram

2) Scenario B:

Controlling: In scenario B, the medical staff remotely controls the healthcare treatment device and the signal flow is shown in Fig. 7. The manager can register a topic resource for a 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.

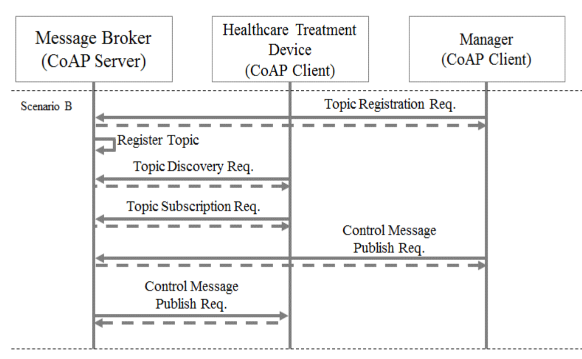


Fig 7. Scenario B: Controlling signal flow diagram

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

MQTT has methods; connect, Disconnect, Subscribe and Publish that indicate what actions should be performed on the specific resources. Connect waits for server establishment connection, Disconnect waits for the MQTT to finish any work it must do and session to be disconnected. Subscribe waits for completion of the subscribe or unsubscribe method, Unsubscribe requests the server to unsubscribe the client from

TABLE II. MAPPED MQTT METHODS TO CoAP METHODS USED IN THE PROPOSED MESSAGING SYSTEM

MQTT Functions	CoAP Methods	Explanation
Publish	GET	Resource Discovery, Request Resource Data
Unsubscribe	DELETE	Client unsubscription from a topic
Subscribe	Observe	Request Subscription

one or more topics and Publish returns immediately to the application thread after passing the request to the MQTT client. All these methods were mapped to the CoAP native methods so that it can possess the MQTT methods operations and as described in Table II.

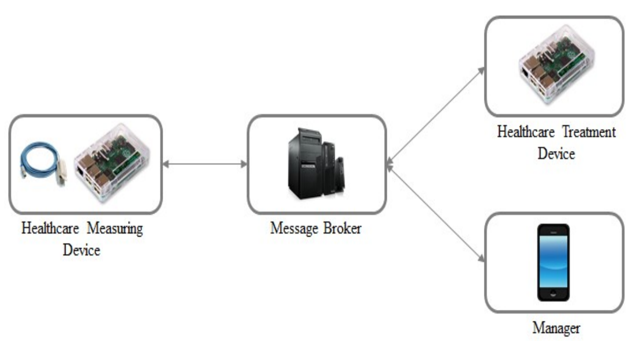
IV. Implementation

In this section, we implemented a prototype of the interoperable messaging system that uses CoAP for IoT healthcare services.

A. *Implementation Configuration*

Fig. 8 illustrates a realistic prototype for our proposed messaging system for IoT healthcare services.

In this experiment, two raspberry pi, spO2 sensor, one android smart phone, one desktop computer and one AP were used. 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. All the different components were connected together.



a) Fig 8. The Messaging System Prototype for IoT healthcare services that uses CoAP.

B. *Implementation Results*

Fig. 9 illustrates a CoAP message broker analysis tool known as CoAP Copper plugin which is an extension program added onto Firefox. ISO/IEEE 11073 DIM was created by referencing the ISO/IEEE 11073-10404 document. ISO/IEEE 11073-10404 document defines pulse oximeter specialization. Pulse rate and oxygen saturation were measured and is illustrated in Fig. 9. Fig 9 also contains message broker address at the top and resource list of the message broker on its left side. Looking at the resource list section, there are topic resources called MeasurementData and ControlData. Just at the center of Fig. 9, we can see a response message from the message broker.

Fig. 10 shows a screen shot of a publisher client. If there is no registered topic resource on the message broker, publisher client requests for a topic resource registration. Thereafter, publisher client publishes ISO/IEEE 11073 DIM message when healthcare data is measured from pulse oximeter sensor and the implementation result of the publisher client is shown in Fig. 10. Fig. 10 and Fig.11 both show subscriber client screen shots implemented on raspberry pi and smartphone respectively.

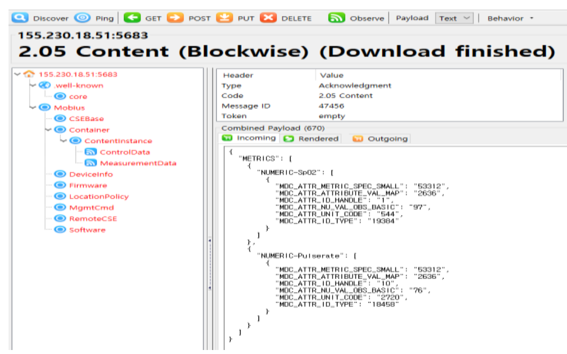


Fig 9. CoAP message broker analysis tool

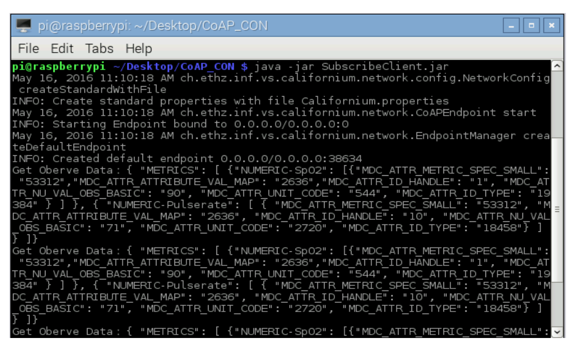


Fig 10. Subscriber client screenshot implemented on a raspberry pi.

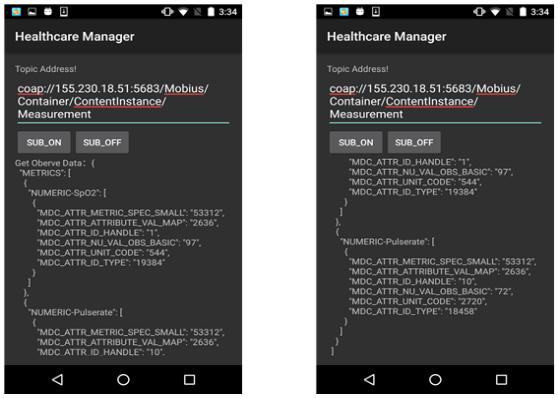


Fig 11. Subscriber screenshot implemented on a smartphone.

V. Performance Evaluation

For evaluation of the proposed design, we compare and analyze the CoAP based messaging system against the MQTT publish/subscribe messaging system. This performance test was executed to validate the key features of the proposed messaging system, such as light weight property, reliability provision, etc. In this simulation environment, the loss rate of 5, 10, 15, 20, 25, and 30% were added and analyzed accordingly.

For reliability, CoAP supports reliability by using CON and NON messaging option while MQTT protocol supports reliability using QoS 0, 1, and 2. Fig. 12 illustrates the average RTT (Round Trip Time) in one complete transaction. We compared CoAP CON vs MQTT QoS 1 and 2 for this case. These results are not in any way based on the different performance effects of UDP and TCP. This therefore implies that CoAP RTT at any one point is shorter than MQTT.

Assuming that loss rate was neglected here, Fig. 13 shows the number of packets used when the healthcare measurement device sends the measured biomedical message to the message broker which is transferred to both the treatment device and the manager by the message broker instantly. For a fair comparison, the same biomedical message was transferred while using CoAP and MQTT separately. This was accomplished by installing each protocol's broker and clients on the same desktop computer while using the same Raspberry Pi. We published the information once after every 10 seconds for ten times. Fig. 13 illustrate that CoAP protocol only used 2 packets for each transfer. Transferring by NON option of CoAP, only 1 packet can be used for the transfer, but then reliability falls at the same time. MQTT_SUB client transfers the message 10 times on a single connection. To do this, the 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_PUB client establishes and terminates the connection each time when the message is transferred, using 12 packets for each transfer.

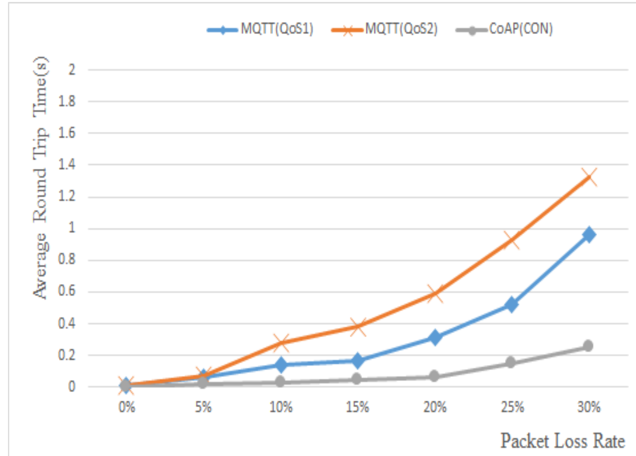


Fig 12. Average RTT for each Packet Loss Rate

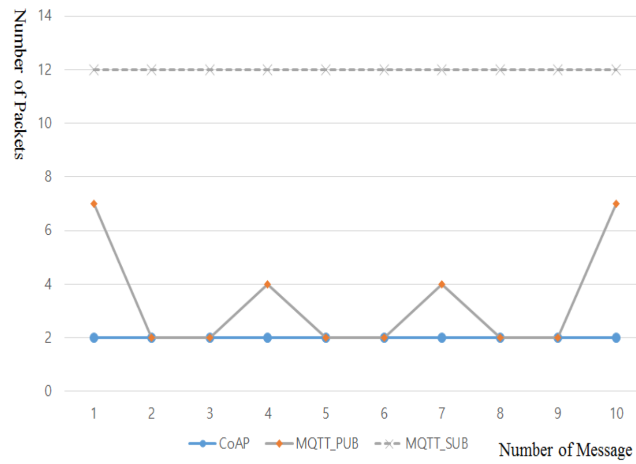


Fig 13. Number of Packets used for transferring data

VI. Conclusion

In this paper, we present a messaging system for IoT healthcare services using CoAP that implements publish/Subscribe messaging model taking in account international standards for IoT Healthcare services. An architecture and a detailed signaling procedures for the messaging system were designed.

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 were analyzed. The results shows that the proposed CoAP based messaging system is superior to previous messaging systems. We hope to extend this design to the other sectors in the IoT.

References

[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] MQTT in Wikipedia http://en.wikipedia.org/wiki/MQTT
[7] Lee, Shinho, et al. "Correlation analysis of MQTT loss and delay according to QoS level." Information Networking (ICOIN), 2013 International Conference on. IEEE, 2013.
[8] Banks, Andrew, and Rahul Gupta. "MQTT Version 3.1.1." OASIS Standard (2014).
[9] Castellani, A., et al. "Guidelines for HTTP-CoAP Mapping Implementations." draft-ietf-core-http-mapping-05 (2014).
[10] Shelby, Z., K. Hartke, and C. Bormann. "The Constrained Application Protocol (CoAP)(RFC 7252)." (2014).