

IoT 기반 저전력 · 경량 프로토콜 표준화 분석

장영환¹ · 심재성¹ · 박석천^{2*}

Analysis Standardized of IoT-based Low-power-Light-weight Protocol

Young-hwan Jang¹ · Jae-sung Shim¹ · Seok-cheon Park^{2*}

¹Department of IT Convergence Engineering, Gachon University, Gyeonggi-do 13120, Korea

²Department of Computer Engineering, Gachon University, Gyeonggi-do 13120, Korea

요 약

최근 스마트 디바이스를 통해 사람과 사물, 사물과 사물이 연결되는 IoT가 미래의 새로운 경제성장 동력으로 부상하고 있다. 이에 국제 표준화 단체에서 IoT 기반 프로토콜에 대한 연구를 진행하고 있으나, 표준화 단체에 소속되어 있는 기관 및 기업들의 이해관계가 서로 상이하고 개발되는 프로토콜 또한 다르기 때문에 기기종 디바이스간의 상호 호환성이 저하되는 문제점이 나타나고 있다. 따라서 본 논문에서는 IoT 환경에 적합한 저전력·경량 프로토콜의 동향을 파악하고자 보편적으로 사용하고 있는 MQTT와 CoAP의 구조, 발전 과정 및 표준화 동향, 특징을 조사 및 분석하여 장단점을 도출 및 비교분석하고, 각 프로토콜에 대한 발전방안을 제시하였다.

ABSTRACT

People and goods through the recent smart device, IoT to things and things are connected, it has emerged as the driving force of a new economic growth in the future. While conducting research of this international standardization organization in the IoT-based protocol, to the institutions and companies that belong to the standards bodies interests are different from each other, protocols have been developed also, different order, heterogeneous device interoperability has emerged is a problem that the lowering of. Therefore, in this paper, to investigate the structure of the MQTT and CoAP that are commonly used and trying to figure out the trend of low power consumption, a lightweight protocol that is suitable for IoT environment, process and standardization trends of evolution, features, analysis and on, to derive the advantages and disadvantages, and comparative analysis, it presented a development scheme for each protocol.

키워드 : 표준화, 사물인터넷, 저전력, 경량, 프로토콜

Key word : Standardization, IoT, Protocol, Low energy, Light weight

Received 28 June 2016, Revised 15 July 2016, Accepted 21 July 2016

* Corresponding Author Seok-Cheon Park(E-mail:scpark@gachon.ac.kr, Tel:+82-31-750-5747)

Department of Computer Engineering, Gachon University, Gyeonggi-do 13120, Korea

Open Access <http://dx.doi.org/10.6109/jkice.2016.20.10.1895>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서론

최근 스마트폰을 포함한 스마트 디바이스의 급속한 확산과 함께 사람과 사물 등 모든 것이 연결되는 IoT(Internet of Things)가 미래의 새로운 경제성장 동력으로 부상하고 있다[1].

이에 IoT 시장을 선점하고자 IETF(Internet Engineering Task Force), OASIS(Organization for the Advancement of Structured Information Standards) 등 표준화 단체를 중심으로 IoT 기반 스마트 디바이스들의 데이터 수집, 메시지 전송 등에 대한 표준 프로토콜 연구가 진행되고 있다[2].

그러나 표준화 단체에 소속되어 있는 기관 및 기업들의 이해관계로 인해 각 단체마다 서로 상이한 표준 프로토콜을 개발하고 있는 상황이다. 또한 제조사에서도 표준화 단체의 프로토콜을 혼용하여 제품을 개발하고 있어, IoT 환경 구축을 위한 기기종 디바이스간의 상호 호환성이 저하되는 문제점이 발생하고 있다.

따라서 본 논문에서는 IoT 환경에서 보편적으로 사용하는 MQTT와 CoAP 프로토콜의 구조, 특징, 표준화 동향을 조사하여 각 프로토콜에 대한 장단점을 도출하였다. 또한 IoT 환경에 적합한 주요 요소를 통해 MQTT와 CoAP 프로토콜 비교분석 및 발전 방안을 제시하였다.

본 논문의 구성은 1장 서론에 이어 2장에서 MQTT의 구조, 발전 과정, 특징을 조사하고, 3장에서는 CoAP의 구조와 표준화 동향, 특징을 조사하였다. 4장에서는 MQTT와 CoAP 프로토콜의 장단점을 도출 및 비교분석하고, 이를 기반으로 발전방안을 제시하였다. 마지막 5장에서는 결론을 기술한다.

II. MQTT 프로토콜

2.1. MQTT의 구조

MQTT(Message Queue Telemetry Transport)는 스마트 디바이스와 같이 대역폭이 제한된 환경에 최적화되어 개발된 푸시 기술 기반의 경량 메시지 전송 프로토콜이다. 일반적인 푸시 기술에 사용되는 클라이언트/서버 방식 대신, 브로커를 통해 송신자가 특정 메시지를 Publish하고, 수신자가 메시지를 Subscribe하는

Publish/Subscribe 방식을 사용한다.

MQTT는 개방적이고 비교적 쉽게 구현할 수 있고, Publish/Subscribe 관계 모델 구조로 설계되었기 때문에 수많은 클라이언트가 하나의 서버에서 지원된다.

이를 통해 M:N 송출이 가능하고, 메시지를 전후방으로 전송함으로써 간결함과 직관성을 보장한다. MQTT의 기본 구조는 그림 1과 같다[3].

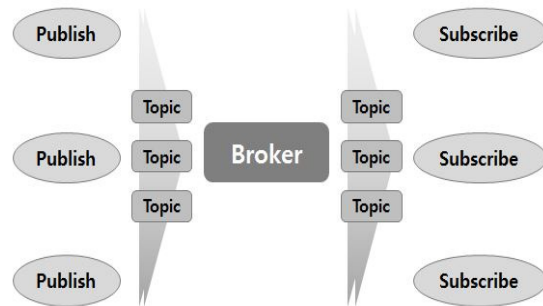


Fig. 1 Basic Structure of MQTT

2.2. MQTT 발전 과정

MQTT는 1999년 IBM에서 통신을 위한 프로토콜로 개발되었으며, 공식 홈페이지를 통해 스펙 및 클라이언트 라이브러리를 제공함으로써, 센서와 모바일 디바이스들의 통신 연결을 위한 프로토콜로 연구가 진행되었다. 이후, 2013년 국제 표준화 단체인 OASIS에서 MQTT를 IoT 표준 프로토콜로 사용하고 있으며, 전기·전자, 통신, 인터넷 등 다양한 분야에 활용될 수 있도록 지속적인 업데이트가 진행되고 있다. MQTT의 발전 과정은 그림 2와 같다[4].

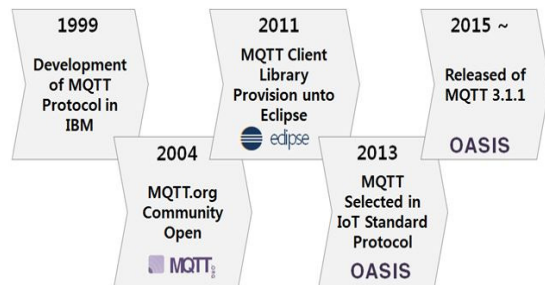


Fig. 2 MQTT Development Process

2.3. MQTT 특징

MQTT는 IoT 환경에 적합한 통신을 위해 경량화, 유연성·확장성, 생산성 등의 특징을 지니고 있다. MQTT의 주요 특징은 표 1과 같다.

Table. 1 MQTT Main Features

Section	Content
Lightweight	- Lightweight Message Format - Memory Level Size of 80~100kb
Flexibility/ Scalability	- Support of Numerous User/Device - Extension is Possible without Changing the Application Code
Productivity	- Detected without Another Application Logic, Save, Provide Functions such as Transfer - A Simple Concept, Easy Learning of Development

경량화는 경량 메시지 포맷을 사용하여 80~100kb 수준의 메모리를 사용하므로 구현이 간단하며, 이벤트 방식과 QoS(Quality of Service)를 통해 다수의 사용자와 디바이스를 지원하고, 응용코드의 변경 없이 기능을 확장할 수 있는 유연성·확장성을 가지고 있다.

또한 생산성 부분은 탐지, 저장, 전달 Publish / Subscribe 기능을 제공하기 때문에 별도의 응용 로직이 필요 없고, 간단한 개념으로 만들어지므로 개발자의 학습이 용이하다.

MQTT는 IoT 통신 환경에 적합하게 개발된 프로토콜로써 디바이스들의 처리 능력, 네트워크 대역폭, 메시지 오버헤드 등 주변 상황에 맞게 시스템이 동작할 수 있도록 3단계의 QoS를 제공하고 있다. 각 단계에 대한 내용은 다음과 같다.

- 0단계 : 메시지는 한번만 전달되며, 전달여부를 확인하지 않는 Fire and Forget 타입이다.
- 1단계 : 메시지는 반드시 한번 이상 전달되지만, 메시지의 핸드셰이킹 과정을 추적하지 않기 때문에 중복 전송될 수 있다.
- 2단계 : 메시지는 한번만 전달되며, 메시지의 핸드셰이킹 과정을 추적하기 때문에 고품질을 보장하지만 다소 높은 성능이 필요하다.

이와 같이 QoS는 3단계로 이루어져 있으며 단계별 장단점을 가지고 있어, IoT 구축 환경에 적합한 단계를 선택하여 사용해야 한다.

먼저 0단계의 경우 오프라인에서의 메시지 박스 서비스 제공 시 다음 연결 메시지를 보내기 위해 자체 큐에 저장하지 않고 읽기 유무를 판단할 필요가 없는 상황에서 사용하기 적합하다.

반면에 No TCP/IP와 TCP/IP가 혼용된 로컬 네트워크에서는 네트워크 구간을 신뢰할 수 없고, 메시지 전송 실패 시 메시지 박스 저장 시스템을 구축하기 어렵기 때문에 1, 2단계를 선택하여 사용하는 것이 효율성을 높일 수 있다. QoS 단계별 구성은 그림 3과 같다[5].

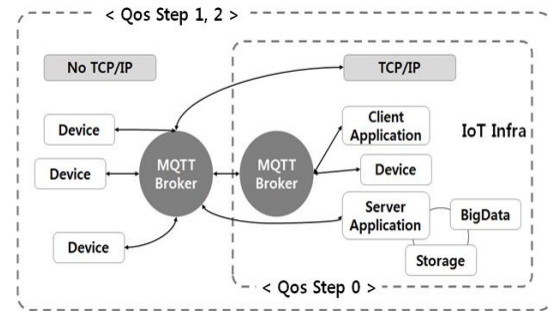


Fig. 3 QoS Configuration of MQTT

III. CoAP 프로토콜

3.1. CoAP의 구조

CoAP은 소형 디바이스들의 인터넷 통신을 지원하기 위해 IETF에서 개발된 프로토콜로, 저전력 센서, 스위치 등을 표준화된 인터넷 환경에서 사용하기 위한 목적을 지니고 있다.

CoAP은 Request/Response 모델로 동작하며, 메시지의 반복전송을 통해 신뢰성을 확보한다.

또한 CoAP 메시지는 2진 포맷으로 인코딩되고, 타입-길이-값(TLV) 포맷을 따라 헤더가 고정된 크기를 가지고 생성되며, 프레임 내의 베이스 헤더는 버전, 메시지 타입, 토큰 길이, 메시지 종류 코드 등으로 구성되어 있다.

CoAP 프로토콜 전송계층은 UDP와 애플리케이션

사이에 위치하고 있으며, 두 가지 레이어로 정의 할 수 있다. CoAP 기본 구조는 그림 4와 같다[6].

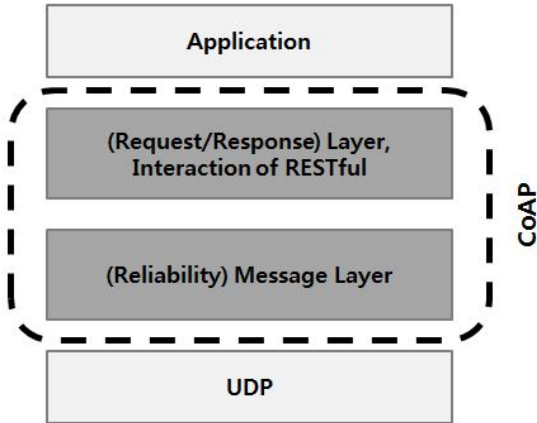


Fig. 4 Basic Structure of the CoAP

3.2. CoAP 표준화 동향

CoAP 프로토콜은 인터넷 표준 단체인 IETF에서 다양한 디바이스가 인터넷에 연결될 것으로 예상하고, 워킹그룹을 만들어 소형 디바이스에 탑재되는 표준을 연구하면서 시작되었다.

워킹그룹 중 하나인 CoRE(Constrained RESTful Environments)에서 소용량/저성능 노드를 손실이 높고 전송률이 낮은 네트워크에서 경량화된 방식으로 메시지를 주고받을 수 있는 CoAP 프로토콜을 개발하였다.

이후, IETF에서 표준 프로토콜로 사용하고 있으며, IoT 표준화 단체인 oneM2M, OIC(Open Interconnect Consortium)에서도 CoAP을 기반으로 IoT 환경 구축을 위한 표준화를 진행하고 있다.

oneM2M에서는 CoAP 프로토콜을 기반으로 IoT 응용 서비스를 지원해주는 공통 서비스 플랫폼과 이와 상호작용하기 위한 인터페이스를 정의하여 새로운 버전의 표준 플랫폼을 발표하였다. 또한 OIC는 CoAP을 기반으로 IoTivity라는 오픈소스 프로젝트를 개발하였으며, 디바이스별 CoAP의 리소스 타입, 탐색 절차, 보안 등의 규격을 정의하였다.

최근 CoAP 상호운용성 시험을 통해 디바이스 간 상호연동이 원활하게 이루어져 IoT 환경에 적합한 통신 프로토콜로 활용되고 있다. CoAP을 기반으로 사용하는 주요 표준화 단체의 동향은 그림 5와 같다[7].

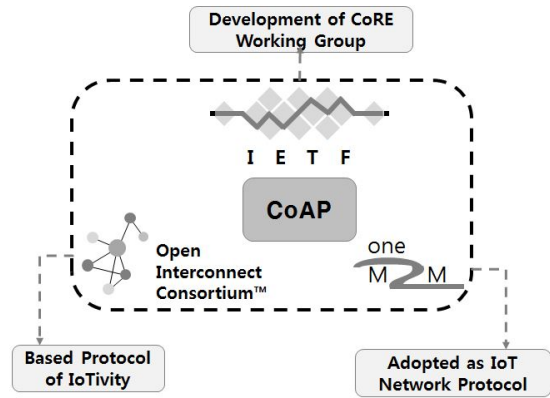


Fig. 5 CoAP Base of Standardization Trends

3.3. CoAP 특징

CoAP 프로토콜은 UDP 기반으로 메시지 교환을 통해 신뢰성/비신뢰성 메시지 전송이 가능하다. CoAP의 메시지 전송 방법은 그림 6과 같다.

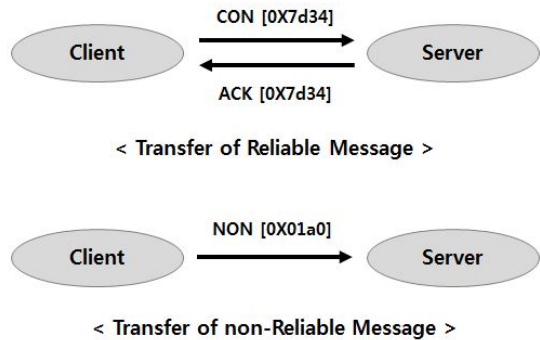


Fig. 6 Transfer Method Message of CoAP

또한 CoAP은 CON(Confirmable), ACK(Acknowledgement), NON (Non-confirmable), RST(Reset)의 4가지 메시지 타입을 정의하여 메시지 전송을 지원한다. 정의하는 내용은 다음과 같다[8].

- CON : ACK를 요청하는 메시지로써, 동기적/비동기적 메시지로 사용가능하다.
- ACK : 응답하지 않아도 되는 메시지이다.
- NON : 확인된 메시지에 대한 응답을 한다.
- RST : 처리될 수 없는 메시지를 확인한다.

또한 CoAP 프로토콜은 HTTP의 명령어 방식을 따르는 RESTful(Representational State Transfer)과 동일한 Request/ Response 방식으로써, 매핑을 통해 기존의 HTTP 프로토콜과도 쉽게 변환 및 연동이 가능하다. HTTP와의 연동 방식은 그림 7과 같다.

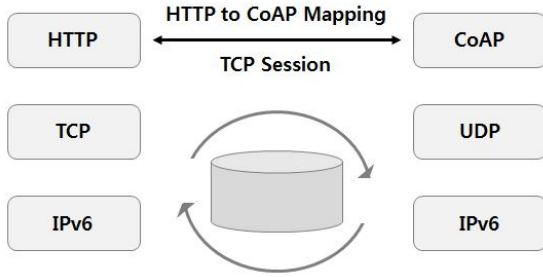


Fig. 7 CoAP and HTTP Linkage Method

CoAP은 기본적으로 IPv6를 기반으로 구축되어, 멀티캐스트 및 유니캐스트가 내재되어 있으며, Observe 모드를 통해 노드가 실제 메시지 내에 참여하지 않으면서도 다른 노드를 관찰할 수 있다[9].

IV. MQTT, CoAP 비교분석

4.1. MQTT 및 CoAP 장단점 도출

4.1.1. MQTT의 장단점

MQTT는 TCP 기반으로 동작하는 프로토콜로써, Publish/Subscribe 관계 모델 브로커에 연결하여 토픽을 기준으로 작동한다. 토픽은 슬래시를 이용하여 계층적인 노드 구성이 가능하므로 다수의 센서 디바이스들을 효율적으로 관리할 수 있는 장점이 있다. 또한 MQTT는 브로커라는 중앙 집중형 서버를 통해 메시지를 전후방으로 전송할 수 있어 간결함과 직관성을 보장하는 장점을 가진다.

MQTT는 브로커라는 서버와 지속적인 연결을 보장하기 때문에 NAT(Network Address Translation) 환경에서 이상 없이 동작한다는 장점이 있다. 그러나 MQTT는 서비스 발견 기능이 없기 때문에 SSDP(Simple Service Discovery Protocol) 등의 추가 사용이 필요하다. MQTT의 장단점은 표 2와 같다.

Table. 2 Advantages and Disadvantages of MQTT

	Advantages	Disadvantages
MQTT	<ul style="list-style-type: none"> - Applied of Publish/Subscribe Relationship Model - Simplicity·Intuitive Guarantee - Operable Possible of NAT Environment 	<ul style="list-style-type: none"> - Absence of a Service Discovery Function - Requires Additional use of SSDP

4.1.2. CoAP의 장단점

CoAP은 기본적으로 UDP에서 실행되며, 일관된 연결 대신 반복적인 메시징에 의존하여 신뢰성을 제공한다는 장점이 있다.

예를 들면, 온도 센서는 온도 변화가 없어도 지속적으로 온도를 체크하여 업데이트하므로 수신 노드가 하나의 정보를 놓치더라도 다음 정보가 수 초 이내로 도착하게 된다.

CoAP은 기본적으로 분산적인 패킷 형태로 이루어져 있어 분산 환경에서의 디바이스나 네트워크 상태정보에 사용할 수 있는 것이 장점이다. 또한 Observe를 통해 서비스를 발견, 관찰할 수 있는 방법을 기본 제공함으로써, 노드가 실제로 참여하지 않으면서 다른 노드를 관찰할 수 있는 장점이 있다.

CoAP은 클라이언트/서버 기반으로, HTTP의 GET, POST, PUT, DELETE 메소드를 이용할 수 있으며, HTTP의 네트워크 연동방식을 이용하는 RESTful과도 상호 운용이 가능한 장점을 지니고 있다. 그러나 CoAP은 UDP 패킷만을 활용함으로써, NAT 환경에서 동작시키기 위해서는 터널링을 하거나, 포트 포워딩과 같은 방법을 사용해야만 하는 단점이 있다. CoAP의 장단점은 표 3과 같다.

Table. 3 Advantages and Disadvantages of CoAP

	Advantages	Disadvantages
CoAP	<ul style="list-style-type: none"> - Suitable to the Status Information in a Distributed Environment - Basic Provides a Service Discovery Function - HTTP, RESTful and Interoperable 	<ul style="list-style-type: none"> - Requires Additional Features for Operation in a NAT Environment

4.2. MQTT, CoAP 비교분석

본 논문에서는 MQTT와 CoAP의 장단점을 기반으로 IoT 환경에 적합한 주요 요소인 프로토콜, 통신노드, 전력소모, 보안 등을 중심으로 비교분석하고, MQTT와 CoAP에 대한 발전 방안을 제시하였다.

4.2.1. MQTT와 CoAP 비교분석 결과

MQTT는 TCP를 기반 프로토콜로 사용하고 있으며 브로커를 통해 완고한 연결을 맺기 때문에 신뢰성 있는 메시지 전송이 가능한 반면, CoAP은 신뢰성 보장이 어려운 UDP 기반에서 동작하기 때문에 반복 전송 등 추가적인 요소를 통해 신뢰성을 확보한다.

통신노드 부분에서는 MQTT는 M:N 방식으로써 한번에 다수의 장치와 연결을 지원하지만 CoAP은 기본적으로 1:1 방식을 지원함으로써, 한 번에 하나의 통신만을 지원할 수 있다.

전력소모는 MQTT가 Publish/Subscribe와 브로커를 이용하는 프로토콜로 Subscribe나 브로커의 요구사항에 따라 토픽을 Publish하기 때문에 특별한 요구사항이 없는 CoAP에 비해 다소 높은 전력을 사용한다.

보안성의 경우 MQTT는 TCP 기반에서 사용되는 TLS, SSL 기반으로 데이터 변조, 도청 등을 방지한다.

CoAP은 UDP 기반에서 사용할 수 있는 DTLS를 통해 TLS와 같이 보안성을 제공하지만, IoT 환경에서의 사용을 고려하지 않아 다소 무거운 보안 기법이다.

마지막으로 MQTT는 네트워크 환경에 따라 선택하여 사용할 수 있는 3단계의 QoS를 지원하고 있으나, CoAP은 QoS를 기본적으로 지원하고 있지 않으며, 별도의 게이트웨이를 추가 및 설치해야 지원이 가능하다. MQTT와 CoAP의 비교분석은 표 4와 같다.

Table. 4 Comparison Analysis

Section	MQTT	CoAP
Based Protocol	TCP	UDP
Communication Node	M:N	1:1
Power Consumption	Relatively High	Relatively Low
Security	TLS, SSL	DTLS
QoS	Self Support	Need to be Separately Implemented

4.2.2. MQTT와 CoAP의 발전방안

MQTT는 비교적 간단한 구조와 역할로 이루어진 TCP 기반의 프로토콜로써, 대역폭이 작은 네트워크에서 동작이 용이하므로, 웨어러블 디바이스와 같이 넓은 대역폭을 요구하지 않는 소형 디바이스에 활용할 수 있다. 또한 브로커 서버를 통해 메시지나 데이터를 전송하는 중앙 집중형으로 구성되어 있어 브로커 서버가 정전, 트래픽 공격 등으로 인해 다운될 경우, 이에 대한 대응방안이 미흡하므로 주기적인 백업, 가상화 서버 등을 통해 문제점을 개선해야 한다.

그리고 MQTT는 유동적인 네트워크 환경에서 적용되는 메시지의 QoS에 따라 패킷 손실과 딜레이가 발생할 수 있기 때문에 QoS의 단계를 확장하거나 하나의 통합된 QoS로 개선해야 한다.

CoAP은 반복적인 전송을 통해 신뢰성을 확보 할 수 있기 때문에 지하철, 버스 정거장 등의 도착 예정 안내 시스템이나, 반복적인 메시지 전송을 통해 실시간 업데이트가 가능하므로 위치 정보, 날씨 정보 등 실시간성이 중요한 곳에 활용하는 것이 바람직하다.

그러나 CoAP에서 사용하는 DTLS의 경우 기반 암호기법인 AES(Advanced Encryption Standard)가 IoT 환경에서는 다소 무겁기 때문에 AES보다 경량화 된 암호화 기법을 이용해서 보안성을 개선해야 한다.

V. 결론

최근 스마트 디바이스의 급속한 확산에 따라 사람과 사물 등 모든 것이 연결되는 IoT가 미래 경제성장 동력으로 주목받고 있다.

이에 IoT 시장 선점을 위해 표준화 단체를 중심으로 IoT 기반 스마트 디바이스들의 데이터 연결, 수집, 전송 등에 대한 표준 프로토콜 연구가 진행되고 있다.

그러나 표준화 단체에 소속되어 있는 기관 및 기업들 간의 이해관계로 인해 각 단체마다 서로 다른 프로토콜을 개발하고 있는 상황이다.

또한 제조사에서도 표준 프로토콜을 혼용하여 제품을 개발하므로 IoT 환경 구축을 위한 이기종 디바이스 간 상호호환성이 저하되는 문제점이 발생하고 있다.

따라서 본 논문에서는 IoT 환경에 적합한 주요 요소에 따라 표준화 단체에서 개발된 MQTT와 CoAP의 구

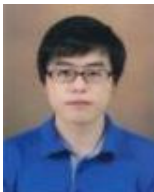
조, 특징, 발전 과정 및 표준화 동향을 조사 및 분석하고, MQTT와 CoAP 프로토콜에 대한 장단점 도출 및 비교 분석을 통해 발전방안을 제시하였다.

ACKNOWLEDGMENTS

This Research was Supported by Basic Science Research Program Through the National Research Foundation of Korea(NRF) Funded by the Ministry of Education(No. 2015R1D1A1A01060184)

REFERENCES

- [1] H. Ahn, Y. J. Lee, K. H. Kim, "A Process-driven IoT-object Collaboration Model," *Journal of Internet Computing and Services*, vol.15, no.5, pp. 9-16, Oct. 2014.
- [2] C. M. Kim, M. G. Kang, "Standard and Trend Analysis of IoT/OneM2M," *Review of Korean Society for Internet Information*, vol.15, no.2, pp. 31-36, Dec. 2014.
- [3] H. Jung, C. W. Park, "Design and Implementation of MQTT Based Real-time HVAC Control Systems," *Journal of the Korea Institute of Information and Communication Engineering*, vol.19, no.5, pp. 1163-1172, May 2015.
- [4] D. K. Lee, D. J. Choi, "Implementation of Zigbee- based Publish/Subscribe System for M2M/IoT Services," *Journal of Korea Multimedia Society*, vol.17, no.12, pp. 1461-1472, Dec. 2014.
- [5] S. H. Kim, D. H. Kim, H. S. Oh, H. S. Jeon, H. J. Park, "The Data Collection Solution Based on MQTT for Stable IoT Platforms," *Journal of the Korea Institute of Information and Communication Engineering*, vol.20, no.4, pp. 728-738, Apr. 2016.
- [6] S. K. Ko, I. K. Park, S. C. Soon, B. T. Lee, "Trends of IETF CoAP Based Sensor Connection Protocol Technology," *Electronics and Telecommunications Trends*, vol.28, no.6, pp. 133-140, Dec. 2013.
- [7] S. K. Ko, B. T. Lee, K. W. Park, "Standardization and Testing of the Trend of CoAP Protocol for the Internet of Things," *TTA Journal*, vol.154, pp. 84-88, 2014.
- [8] Y. J. Park, D. K. Min, "Research on Service-Level Integration of ETSI M2M Platform with Heterogeneous Web Applications based on WSDL 2.0," *Journal of the Korea Institute of Information and Communication Engineering*, vol.17, no.8, pp. 1842-1850, Aug. 2013.
- [9] L. Hang, W. Q. Jin, D. H. Kim, "A Design and Implementation for Registration Service of IoT Embedded Node using CoAP Protocol-based Resource Directory in Mobile Internet Environments," *Journal of the Institute of Internet, Broadcasting and Communication*, vol.16, no.1, pp. 147-153, Fed. 2016.



장영환(Young-Hwan Jang)

2015년 : 평생교육진흥원 멀티미디어학(공학사)
 2015년 ~ 현재 : 가천대학교 IT융합공학과 컴퓨터공학 석사과정
 ※관심분야 : 사물인터넷, 통신 프로토콜, 데이터 관리



심재성(Jae-Sung Shim)

2011년 : 평생교육진흥원 컴퓨터공학(공학사)
 2013년 : 가천대학교 전자계산학과(석사)
 2014년 ~ 현재 : 가천대학교 IT융합공학과 컴퓨터공학 박사과정
 ※관심분야 : 네트워크, 사물인터넷, 표준화



박석천(Seok-Cheon Park)

1977년 : 고려대학교 전자공학과(공학사)
1982년 : 고려대학교 컴퓨터공학(공학석사)
1989년 : 고려대학교 컴퓨터공학(공학박사)
1988년 ~ 현재 : 가천대학교 IT대학 컴퓨터공학과 정교수
※관심분야 : 사물인터넷, 통신 프로토콜, 표준화