

대규모 사물인터넷을 위한 그룹 토픽 기반 MQTT 구독 기법 연구

서윤찬¹, 권정혁^{2*}, 김의직^{3*}

¹한림대학교 소프트웨어학부 학사과정, ²한림대학교 스마트컴퓨팅연구소 연구교수, ³한림대학교 소프트웨어학부 교수

Group Topic-Based MQTT Subscription for Large-Scale Internet of Things

Yun-Chan Seo¹, Jung-Hyok Kwon^{2*}, Eui-Jik Kim^{3*}

¹B.S. Course, Division of Software, Hallym University

²Research Professor, Smart Computing Laboratory, Hallym University

³Professor, Division of Software, Hallym University

요약 본 논문은 수많은 센서가 서로 다른 토픽 데이터를 생성하는 대규모 사물인터넷(Internet of Things, IoT) 환경에서, 다중 토픽 구독의 비효율을 개선하기 위한 그룹 토픽 기반 Message Queuing Telemetry Transport (MQTT) 구독 기법을 제안한다. 기존 MQTT는 브로커가 방대한 개별 토픽 구독 상태를 독립적으로 관리해야 하고, 구독 요청 시 모든 필터 문자열을 포함해야 하므로 메시지 크기와 브로커 처리 부하가 증가하는 구조적 한계를 가진다. 이를 개선하기 위해 제안한 기법은 브로커 내부에 그룹 토픽 생성기 모듈을 도입한다. 그룹 토픽 생성기는 구독자의 요청에 따라 다수의 토픽을 하나의 논리적 그룹 토픽으로 통합하고 맵핑 정보를 관리한다. 이를 통해, 구독자는 다수의 개별 토픽 리스트 대신 단일 그룹 토픽만을 활용하여 다중 토픽 구독을 요청하며, 브로커는 그룹 단위 세션 관리로 구독 상태 관리를 단순화한다. 성능 평가 결과, 제안된 방식이 기존 MQTT 대비 구독 지연과 브로커 메모리 사용량 측면의 성능을 크게 향상하게 시키는 것으로 나타났다.

주제어 : 그룹 토픽, 다중 토픽 구독, 대규모 사물인터넷, 멀티 모듈 브로커, MQTT

Abstract This paper presents a group topic-based Message Queuing Telemetry Transport (MQTT) subscription to improve the efficiency of multi-topic subscriptions in large-scale Internet of Things (IoT) environments, where numerous sensors generate diverse topic data. Conventional MQTT requires the broker to independently manage a large number of individual topic subscriptions and include all filter strings in subscription requests, leading to increased message size and broker processing load. To address these problems, the proposed scheme introduces a Group Topic Generator (GTG) module within the broker. The GTG consolidates multiple requested topics into a single logical group topic and manages the associated mapping information. As a result, the subscribers request multi-topic subscriptions using only a single group topic instead of multiple individual topic lists. Meanwhile, the broker simplifies subscription management through group-based session handling. Performance evaluation demonstrated that the proposed scheme significantly reduces subscription delay and broker memory usage compared to conventional MQTT.

Key Words : Group Topic, Multi-Topic Subscription, Large-Scale Internet of Things, Multi-Module Broker, MQTT

본 논문은 2025년도 교육부 및 강원특별자치도의 재원으로 강원RISE센터의 지원을 받아 수행된 지역혁신중심 대학지원체계(RISE)의 결과입니다(2025-RISE-10-009). 이 논문은 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(RS-2024-00353366).

*교신저자 : 김의직(ejkim32@hallym.ac.kr), 권정혁(jhkwon@hallym.ac.kr)

접수일 2025년 11월 05일 수정일 2025년 11월 28일 심사완료일 2025년 12월 10일

1. 서론

최근 사물인터넷(Internet of Things, IoT) 기술은 스마트홈, 스마트시티, 산업 자동화 등 다양한 영역으로 빠르게 확산되고 있다. 이처럼 IoT가 생활과 산업 전반에 깊이 스며들면서, 수많은 센서와 디바이스가 네트워크를 통해 서로 연결되고 있다. 각 디바이스는 주변 환경이나 사용자 상태를 지속적으로 감지하여 실시간으로 데이터를 주고받으며, 그 결과 생성되는 데이터의 양과 종류가 기하급수적으로 늘어나고 있다 [1, 2]. 이러한 변화로 인해 IoT 시스템은 점차 대규모 네트워크 형태로 발전하고 있으며, 그 안에서 발생하는 방대한 데이터를 지연 없이 처리하고 안정적으로 전달하기 위한 고신뢰성 및 고효율 통신 구조의 필요성이 부각되고 있다 [3, 4].

이러한 요구를 충족하기 위한 대표적인 프로토콜로 Message Queuing Telemetry Transport(MQTT)가 널리 활용되고 있다. MQTT는 발행자(Publisher), 브로커(Broker), 구독자(Subscriber)로 구성된 발행-구독(pub/sub) 구조를 기반으로 하며, 단순한 구조와 낮은 전송 오버헤드를 갖는다 [5-7]. MQTT에서 메시지는 슬래시(/)를 구분자로 하는 계층적 토픽 구조를 통해 전달된다. 구독자는 자신이 원하는 토픽을 정확히 지정하거나, 와일드카드를 사용하여 여러 토픽을 한 번에 구독할 수 있다.

그러나 기존 MQTT는 엄격하게 정의된 계층적 토픽 구조를 따르므로, 다수의 구독자가 서로 다른 토픽을 구독하는 대규모 IoT 환경에서는 브로커의 처리 부담과 네트워크 트래픽이 급격히 커지는 한계를 갖는다 [8, 9]. 구체적으로, 브로커는 각 토픽별로 구독자 목록을 개별적으로 관리해야 하므로, 구독 요청과 메시지 전송 시 구독자-토픽 매칭에 필요한 처리 부하가 상승한다 [10]. 또한, 다수의 토픽을 구독하려는 단일 구독자의 경우, 요청 메시지에 구독할 모든 토픽을 포함해야 하므로, 메시지 크기 비대화로 인해 네트워크 트래픽 및 전송 지연의 심화를 초래한다 [11].

이러한 문제를 해결하기 위해 다양한 연구들이 수행되어 왔으며, 이들은 주로 기존 MQTT의 성능 개선, 특히 브로커의 처리 부담을 줄이는 것에 초점을 두고 있다. An et al.은 여러 브로커가 함께 트래픽을 처리하도록 설계하여 단일 브로커의 과부하를 완화했지만, 브로커 간 상태 동기화가 복잡해지는 문제와 구독자의 다중 토픽 관리 문제는 여전히 해결되지 않은 채 남아 있었다 [12]. Zakutynskyi et al.은 CPU 사용률과 메시지 큐

상태 등을 기반으로 부하를 재분배하는 방식을 제안했으나, 지속적인 모니터링에 따른 오버헤드와 복잡한 계산 절차가 단점으로 지적되었다 [13]. Lee et al.은 네트워크 상황에 따라 메시지를 효율적으로 분배하는 공유 구독 기반 정책을 제안했지만, 구독자가 많은 개별 토픽을 구독할 때 발생하는 메시지 비대화와 전송 지연 문제를 해결하지는 못했다 [14].

본 논문에서는 그룹 토픽 기반 MQTT 구독 기법을 제안하며, 대규모 IoT 환경에서 다중 토픽 구독 효율을 극대화하는 것을 목표로 한다. 제안한 기법에서, 브로커는 그룹 토픽 생성기(Group Topic Generator, GTG) 모듈을 내부에 포함하며, 이는 다수의 개별 토픽을 하나의 논리적 그룹 토픽으로 통합하는 기능을 수행한다. 구독자는 구독할 다수의 개별 토픽을 GTG에 전달하여 그룹 토픽을 생성한 뒤, 해당 그룹 토픽을 활용하여 한 번의 요청으로 다수의 토픽을 구독한다. 이 과정에서 브로커는 그룹 토픽을 기준으로 구독 세션을 관리한다. 즉, 구독자와 개별 토픽 각각에 대한 다수의 세션을 생성하는 대신, 구독자-그룹 토픽 간 최소한의 세션만 유지한다. 제안한 기법의 우수성을 확인하기 위해, 실험적 구현을 통한 성능평가를 수행했다. 그 결과, 토픽 수 100개를 기준으로 기존 MQTT 대비 구독 지연 22.7% 단축과 브로커 메모리 부담 97.9% 감소를 달성했다.

본 논문의 구성은 다음과 같다. 제2장에서는 관련 연구에 관해 기술하고, 제3장에서는 제안한 기법에 대해 자세히 설명한다. 제4장에서는 구현 및 실험 결과를 제시하고, 마지막으로 제5장에서 본 논문의 결론을 맺는다.

2. 관련 연구

2.1 기존 MQTT 구독 방식의 구조적 제약

MQTT는 경량 Pub/Sub 프로토콜이지만, 대규모 IoT 환경에서는 구독 구조의 한계로 인해 확장성이 제한된다. 기존 MQTT에서는 M개의 토픽을 구독하기 위해 M개의 필터 문자열을 포함한 구독 요청 메시지(i.e., SUBSCRIBE 메시지)를 전송해야 하며, 이는 메시지 크기 증가와 전송 지연으로 이어진다. 와일드카드("#", "+")를 사용하면 일부 다중 토픽 구독을 단순화할 수 있지만, 서로 다른 상위 계층에 속한 토픽을 동시에 구독할 경우 여전히 여러 필터가 필요하다. 예를 들어 'sensor/room1/#'와 'device/temperature/#'를 함께 구독하려면 두 개의 필터 문자열을 포함해야 한다.

브로커는 각 필터에 대해 개별 매칭과 세션 관리를 수행해야 하므로, 구독자 수와 토픽 수가 증가할수록 처리 부하가 급격히 증가한다. 특히 N명의 구독자가 각각 M개의 토픽을 요청하면 N×M개의 구독 세션을 생성해야 하므로 메모리 사용량과 연산 부담이 크게 누적된다. 이러한 구조적 한계는 대규모 IoT 환경에서 MQTT의 실시간성 및 확장성을 저해하는 주요 요인으로 작용한다.

2.2 MQTT 브로커 성능 개선 관련 기존 연구

An et al.은 Routing Protocol for Low-power and Lossy Networks(RPL) 기반 분산 MQTT 브로커 구조를 설계하여, 단일 브로커에 집중되는 트래픽 부하를 분산하고 시스템 안정성을 높였다 [12]. 그러나, 이러한 접근법은 브로커 간 메시지 동기화와 백업 과정에서 구조적 복잡성을 증가시킨다. 또한, 구독자가 다수의 개별 토픽을 동시에 구독하는 대규모 IoT 환경에서, 브로커가 개별 토픽별로 구독 세션을 관리해야 하므로 메모리 및 처리 부담이 증가한다.

Zakutynskyi et al.은 분산 IoT 환경에서 브로커 간 부하 불균형을 해소하기 위해 CPU 사용률, 메시지 큐 상태, 네트워크 지연 등을 종합적으로 고려한 부하 분산 기법을 제안했다 [13]. 이 접근법은 처리 효율을 높일 수 있지만, 지속적인 모니터링으로 인한 오버헤드와 복잡한 부하 계산 절차라는 단점을 가진다. 또한, 구독자가 개별 토픽을 직접 관리하는 구조를 사용하므로, 브로커에서 발생하는 다중 구독 세션 관리 문제는 여전히 해결되지 못했다.

Lee et al.은 MQTT 5.0의 공유 구독 기능과 Markov Decision Process(MDP) 기반 전송 정책을 적용하여 브로커의 전송 효율을 개선하였다 [14]. 기존 라운드 로빈 방식의 고정적 메시지 분배 구조를 보완하고, 네트워크 상태에 따라 최적 전송 경로를 선택하도록 설계했다. 그러나 구독자가 대량의 개별 토픽을 구독할 때 발생하는 SUBSCRIBE 메시지의 크기 증가로 인한 전송 지연 문제가 존재한다.

이처럼 기존 연구들은 브로커 측면의 효율성 개선에 초점을 맞추었으나, SUBSCRIBE 메시지 크기 증가로 인한 네트워크 측면의 성능 저하와 브로커의 다중 구독 세션 관리 부담이라는 구조적 문제는 해결하지 못하였다. 따라서 대규모 IoT 환경에서 구독 효율성과 브로커 성능을 동시에 확보할 수 있는 새로운 접근법이 필요하다.

3. 그룹 토픽 기반 MQTT 구독 기법

본 장에서는 제안하는 그룹 토픽 기반 MQTT 구독 기법에 대해 자세히 설명한다. 먼저, 그룹 토픽의 발행 및 관리를 위해 멀티 모듈 구조를 갖는 브로커를 제시하고, 이어서 그룹 토픽 발행 및 이를 활용한 구독 절차에 대해 기술한다.

제안한 기법에서 브로커는 개별 토픽 단위로 처리되는 기존의 구독 방식을 대체하며, 이를 논리적인 그룹 단위로 통합 관리하는 핵심적인 연산 주체 역할을 수행한다. 이를 위해 브로커는 다음의 두 가지 내부 처리 모듈을 포함한다.

- **그룹 토픽 생성기(Group Topic Generator, GTG):** 클라이언트의 그룹 생성 요청을 처리하고, 다수의 개별 토픽들을 단일 그룹으로 맵핑 및 관리하는 역할을 담당한다.
- **메시지 처리기(Message Handler, MH):** 기존 MQTT 브로커의 고유 역할인 메시지 수신, 전송 및 라우팅을 담당한다.

이러한 멀티 모듈 구조의 브로커를 통해, 기존 MQTT에서 개별 토픽 단위로 처리되던 구독 및 메시지 전달 과정이 논리적 그룹 단위로 통합된다. 즉, 구독자는 다수의 개별 토픽을 직접 관리할 필요 없이, GTG가 생성한 그룹 토픽을 통해 단일 구독 요청으로 여러 토픽을 동시에 구독하며, MH는 이 그룹 정보를 기반으로 개별 토픽의 메시지를 해당 그룹 구독자에게 정확하게 라우팅한다.

[Fig. 1]은 그룹 토픽의 생성 단계부터, 그룹 토픽 기반 구독 처리에 이르는 전체 동작 절차를 보여주며, 해당 절차는 다음과 같이 총 네 단계로 구성된다.

첫째, 그룹 생성 요청 단계이다. 여러 센서에서 발생하는 서로 다른 토픽의 데이터를 구독하려는 구독자는, 먼저 자신의 토픽 리스트에서 원하는 토픽 조합에 해당하는 그룹 토픽이 이미 존재하는지 확인한다. 그룹 토픽이 존재하지 않으면, 구독자는 구독 대상 토픽 목록을 포함한 그룹 생성 요청 메시지를 GTG로 전송한다. 반대로, 해당 그룹 토픽이 이미 생성되어 있는 경우에는 그 그룹 토픽을 포함한 SUBSCRIBE 메시지를 MH로 전송하여 그룹 토픽 기반 구독을 요청한다.

둘째, 요청 처리 및 그룹 토픽 생성 단계이다. GTG가 그룹 생성 요청 메시지를 수신하면, 요청 메시지에 포함된 토픽 리스트를 확인하고, 해당 조합으로 구성된 그룹 토픽이 이미 존재하는지 확인한다. 만약 해당 그룹 토픽이 존재하지 않는다면, 새로운 그룹 토픽을 생성하여 토

픽 리스트와 맵핑한 후 등록 및 유지한다. 반대의 경우, 구독자에게 해당 그룹 토픽을 회신한다.

셋째, 그룹 토픽 응답 및 공지 단계이다. 생성된 그룹 토픽은 요청을 보낸 당사자뿐만 아니라, 연결된 구독자들에게도 공유되어야 한다. 따라서 GTG는 요청 구독자에게 그룹 토픽을 포함한 응답 메시지를 반환하고, 새롭게 생성된 그룹 토픽의 경우에는 연결된 모든 구독자에게도 공지한다.

마지막으로, 그룹 토픽 기반 구독 단계이다. 그룹 토픽을 전달받은 구독자는 여러 개의 개별 토픽을 SUBSCRIBE 메시지에 나열하는 대신, 해당 그룹 토픽 하나만 포함하여 구독 요청을 수행한다. 브로커가 SUBSCRIBE 메시지를 수신하면 그룹 토픽을 기준으로 단일 구독 세션을 생성한다. 이후 브로커는 이 세션에 해당 그룹에 속한 개별 토픽들의 맵핑 정보를 설정하여, 그룹 단위로 메시지를 전달할 수 있도록 내부 메모리 구조를 구성한다. 이후, SUBACK을 반환하여 구독 요청이 성공적으로 처리되었음을 알린다.

구독 설정이 완료된 후, 발행자(Publisher)가 개별 토픽으로 데이터를 발행하면, 브로커의 MH는 GTG가 생성한 맵핑 정보를 참조한다. MH는 해당 개별 토픽이 포함된 그룹 토픽을 구독하고 있는 모든 구독자에게 해당 데이터를 전달한다.

4. 구현 및 성능 평가

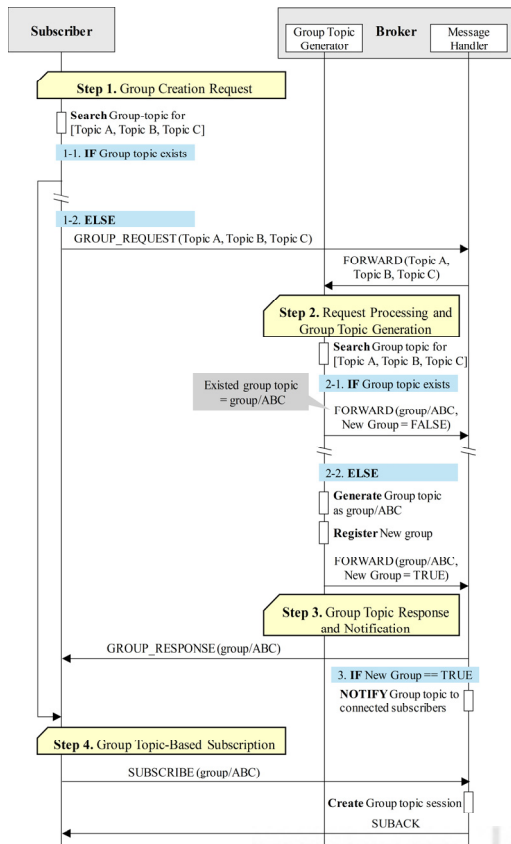
4.1 실험 환경 및 시나리오 구성

실험 환경 구성을 위해, Intel Core i5-1135G7 및 16GB RAM이 탑재된 Windows 11 환경에서 오픈소스 라이브러리를 활용하여 MQTT 시스템의 각 구성요소를 구현하였다. 구체적으로, 브로커의 MH 모듈은 Mosquitto v2.0.21을 기반으로, GTG 모듈은 Node.js v22.15.1을 활용하여 각각 구현하였다 [15]. 브로커의 모듈 간 상호작용과 구독자 및 발행자 운용 소프트웨어는 모두 Node.js를 기반으로 구현하였으며, 외부 요인을 배제하기 위해 동일 머신 내에서 로컬 루프백(Loopback 127.0.0.1)을 통해 상호작용 하도록 구성하였다.

실제 IoT 환경을 고려한 실험 시나리오 구성을 위해, IEEE 802.15.4 표준에서 정의하는 전송률(250kbps)을 적용하였다. 구독자 및 토픽 수 변화에 따른 성능을 평가하고자, 구독자 수가 10명과 20명인 경우로 나누어 실험을 진행하였고, 이때 각 구독자가 수신하는 토픽 수를 10개에서 100개까지 점진적으로 증가시키며 성능을 측정했다. 성능 측정은 Wireshark를 통해 로컬 루프백 환경에서 수행되었다. 제어 메시지의 크기와 발생 시점은 동일하게 Wireshark를 사용하여 분석하였다. 제안한 기법의 우수성을 입증하기 위해, 실험 결과를 기존 MQTT의 결과와 정량적으로 비교하였다. <Table 1>은 실험에서 사용된 주요 파라미터를 보여준다.

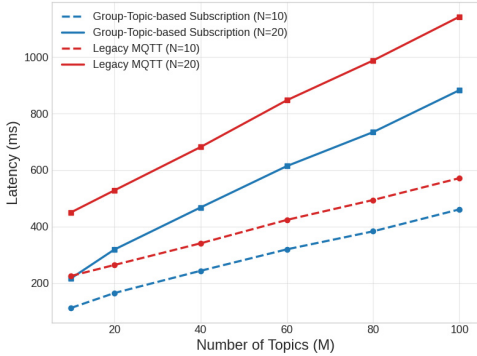
<Table 1> Experiment Parameters

Parameter	Value
Data Rate	250 kbps
Number of Subscribers (N)	10, 20
Number of Topics per Sub (M)	10, 20, 40, 60, 80, 100
MQTT QoS Level	Level 0 (At most once)
Topic String Length	Individual: 7 - 9 Bytes
Group URI:	15 Bytes (sensor/group/g1)
Fixed Control Topic (Request)	31 Bytes
Fixed Control Topic (Info)	20 Bytes

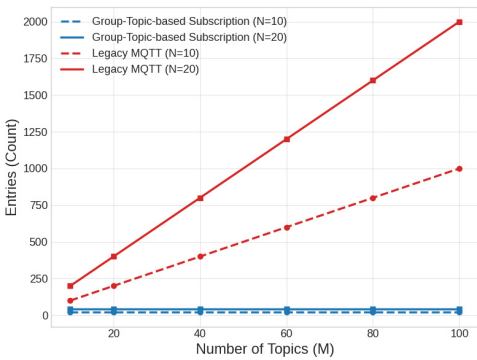


[Fig. 1] Group Topic Generation and Subscription

4.2 실험 결과



[Fig. 2] Subscription Latency



[Fig. 3] Broker Memory Burden

<Table 2> Overhead Time for Group Topic Generation

M	10	20	40	60	80	100
Time (ms)	204.3	305.7	454.9	601.4	721.7	870.2

[Fig. 2]는 토픽 수(M) 증가에 따른 구독 지연 시간 (Latency)의 변화를 보여준다. 제안한 기법은 다수의 토픽을 단일 그룹 토픽으로 통합하여 구독 요청 메시지를 경량화함으로써, 초기 그룹 생성 오버헤드에도 불구하고 기존 MQTT 대비 더 낮은 지연 시간을 달성한다. <Table 2>는 N=20 환경에서 토픽 수 변화에 따른 초기 그룹 생성 오버헤드를 나타낸다. 해당 오버헤드는 토픽 수 증가에 따라 커지지만, 초기 1회만 발생한다. 따라서 반복적인 구독 요청 상황에서는 메시지 크기가 전체 지연 시간에 더 큰 영향을 미친다. 특히 제안한 기법은 M과 무관하게 20 Bytes의 고정 크기 메시지를 전송하는 반면, 기존 MQTT는 토픽 수에 비례해 메시지 크기가 증가한다. N=20, M=100에서 제안한 기법은 약 883 ms, 기존 MQTT는 약 1,143 ms의 지

연 시간을 보였으며, 평균적으로 약 22.7%의 성능 향상을 확인하였다.

[Fig. 3]은 브로커가 관리해야 하는 활성 구독 엔트리 수의 변화를 보여준다. 제안한 기법은 토픽 수의 증가와 무관하게 구독자당 고정된 소수의 엔트리($2N + \alpha$)만을 유지한다. 여기서 $\alpha (=2)$ 는 그룹 생성 요청 및 정보 전달을 위해 시스템이 상시 유지해야 하는 2개의 제어용 고정 토픽을 의미한다. 반면, 기존 방식은 구독자와 토픽 수의 곱($N \times M$)만큼 세션이 생성되어, 최대 부하 시나리오(N=20, M=100)에서 총 2,000개의 엔트리가 메모리를 점유하였다. 실험 결과, 제안한 기법은 동일 조건에서 단 42개의 엔트리($2 \times 20 + 2$)만을 생성하여 기존 대비 97.9%의 메모리 절감 효과를 달성하였다. 이 결과는 대규모 IoT 환경에서 브로커의 자원 효율성을 획기적으로 개선할 수 있음을 보여준다.

5. 결론

본 논문에서는 대규모 IoT 환경에서 MQTT의 구조적 한계인 트래픽 과부하와 브로커 자원 부족 문제를 해결하기 위해, 그룹 토픽 기반 구독 기법을 제안했다. 제안한 기법은 브로커가 다수의 개별 토픽을 하나의 논리적 그룹으로 통합 관리하고, 구독자가 단일 그룹 식별자만으로 구독을 요청하게 함으로써 메시지 페이로드와 관리 엔트리를 획기적으로 최소화한다. 성능 평가 결과, 제안한 기법은 기존 대비 구독 지연 시간을 22.7% 단축하고 브로커의 메모리 부담을 97.9% 감소시키는 성과를 달성하였으며, 이를 통해 대규모 IoT 네트워크에서 요구되는 높은 신뢰성과 통신 효율성을 효과적으로 입증하였다. 향후 연구에서는 실제 적용 환경을 고려한 시뮬레이션 환경을 확장하여, 브로커 자원 사용률과 평균 메시지 크기에 따른 시스템 부하를 정량적으로 분석하고, 다양한 네트워크 트래픽 조건에서 제안한 기법의 안정성과 낮은 오버헤드를 검증할 계획이다.

REFERENCES

- [1] L.Atzori, A.Iera and G.Morabito, "The Internet of Things: A survey," *Comput. Netw.*, Vol.54, No.15, pp.2787-2805, 2010.
- [2] J.Gubbi, R.Buyya, S.Marusic and M.Palaniswami, "Internet of Things (IoT): A vision, architectural

elements, and future directions," *Future Gener. Comput. Syst.*, Vol.29, No.7, pp.1645-1660, 2013.

- [3] H.Moustafa and D.P.Agrawal, "IoT: Security and privacy issues and challenges," *IEEE Internet Things J.*, Vol.3, No.5, pp.724-733, 2016.
- [4] N.H.Thinh, Q.T.Vien, J.B.Kim, H.X.Nguyen and V.T.Pham, "A survey on challenges and solutions for large-scale IoT networks," *J. Netw. Comput. Appl.*, Vol.135, pp.11-25, 2019.
- [5] U.Hunkeler, H.L.Truong and A.Stanford-Clark, "MQTT-S — A publish/subscribe protocol for Wireless Sensor Networks," in *Proc. IPSN*, pp.68-79, 2008.
- [6] A. Stanford-Clark and H. Trigg, "MQTT V3.1.1 Specification," OASIS Standard, 2014.
- [7] N.Al-Saif and S.Al-Rubaye, "A performance evaluation of MQTT and CoAP protocols for Internet of Things applications," in *Proc. ICCA*, pp.252-257, 2017.
- [8] A.Carzaniga, D.S.Rosenblum and A.L.Wolf, "Design and evaluation of a wide-area event notification service," *ACM Trans. Comput. Syst.*, Vol.19, No.3, pp.332-383, 2001.
- [9] P.T.Eugster, P.A.Felber, R.Guerraoui and A.M.Kerमारrec, "The many faces of publish/subscribe," *ACM Comput. Surv.*, Vol.35, No.2, pp.114-133, 2003.
- [10] D.V.Thanh and T.D.Nguyen, "Performance evaluation of MQTT in different deployment scenarios for IoT applications," *J. Sens. Actuator Netw.*, Vol.7, No.2, 2018.
- [11] H.Koziolek, S.Grüner and J.Rückert, "A comparison of MQTT brokers for distributed IoT edge computing," in *Proc. ECSA*, pp. 352-368, 2020.
- [12] H.S.An, W.J.Sa and S.G.Kim, "Design and Implementation of RPL-based Distributed MQTT Broker Architecture," *J. Korea Multimed. Soc.*, Vol.21, No.9, pp.1090-1098, 2018.
- [13] I.Zakutynskyi, I.Robodzei, S.Burmakin, O.Kalishuk and V.Nebylytsia, "Improving a procedure of load balancing in distributed IoT systems," *East.-Eur. J. Enterp. Technol.*, Vol.5, No.2(125), pp.6-22, 2023.
- [14] S.Lee, Y.Kim, Y.Kim, J.Choi and Y.Kyung, "Cost-aware Optimal Transmission Scheme for Shared Subscription in MQTT-based IoT Networks," *J. KICS*, Vol.10, No.4, pp.1-8, 2024.
- [15] Mosquitto.conf Man Page[Internet], <https://mosquitto.org/man/mosquitto-conf-5.html>.

서 윤 찬(Yun-Chan Seo)

[준회원]



- 2021년 3월 ~ 현재 : 한림대학교 소프트웨어학부 학사과정

<관심분야>

사물인터넷, 무선센서네트워크, 임베디드시스템, 머신러닝, 데이터 분석

권 정 혁(Jung-Hyok Kwon)

[정회원]



- 2010년 2월 : 송실대학교 정보통신전자공학부 (공학사)
- 2012년 8월 : 고려대학교 전기전자전파공학과 (공학석사)
- 2019년 2월 : 한림대학교 융합소프트웨어학과 (공학박사)

- 2013년 4월 ~ 2015년 6월 : LG넥스원 SW연구센터 선임연구원
- 2019년 3월 ~ 현재 : 한림대학교 스마트컴퓨팅연구소 연구교수

<관심분야>

사물인터넷, 무선센서네트워크, 무선전력전송, 해상무선통신, 차량 네트워크, 머신러닝, 블록체인

김 의 직(Eui-Jik Kim)

[정회원]



- 2004년 2월 : 고려대학교 전기전자전파공학부 (공학사)
- 2006년 2월 : 고려대학교 전자컴퓨터공학과 (공학석사)
- 2013년 2월 : 고려대학교 전기전자전파공학과 (공학박사)

- 2006년 2월 ~ 2009년 7월 : 삼성전자 DMC연구소 선임연구원
- 2009년 8월 ~ 2013년 8월 : KT 융합기술원 선임연구원
- 2013년 9월 ~ 현재 : 한림대학교 소프트웨어학부 교수

<관심분야>

사물인터넷, 무선센서네트워크, 무선전력전송, 해상무선통신, 차량 네트워크, 머신러닝, 블록체인