

네트워크 상태 기반 MQTT 브로커의 QoS 최적화 전송 방식

정주은¹, 박서연¹, 경연웅^{2*}

¹국립공주대학교 정보통신공학과 학생, ²서울과학기술대학교 전자공학과 교수

Network-Aware QoS Optimization for MQTT Broker-Based Message Delivery

Jueun Jeong¹, Seoyeon Park¹, Yeunwoong Kyung^{2*}

¹Student, Dept. of Information & Communication Engineering, Kongju National University

²Professor, Dept. of Electronic Engineering Seoul National University of Science and Technology

요약 사물인터넷(Internet of Things; IoT) 환경에서 Message Queuing Telemetry Transport(MQTT)는 경량성과 효율성을 바탕으로 널리 활용되는 메시지 전송 프로토콜이다. 그러나 표준 MQTT 구조에서는 구독자가 네트워크 상태를 고려하지 않고 QoS 수준을 고정적으로 설정하기 때문에, 전송 품질 변화에 따라 불필요한 오버헤드가 발생하거나 데이터 손실이 발생할 수 있는 한계가 존재한다. 본 논문에서는 브로커가 실시간으로 네트워크 상태를 평가하고, 구독자의 애플리케이션 클래스가 요구하는 허용 지연 및 손실률을 만족하는 가장 낮은 QoS(Quality of Service) 수준을 동적으로 선택하여 조정한 후 메시지를 전달하는 새로운 QoS 최적화 전송 방식을 제안한다. 제안 방식은 초기 구독시 브로커가 구독자에게 적절한 QoS 수준을 제공하여 설정 오류를 방지하고, 메시지 발행 시점에서 네트워크 상태에 따라 QoS 수준을 재조정함으로써 메시지 전달의 효율성과 신뢰성을 향상시킨다. 수식 기반의 성능 분석 결과, 제안 방식은 특히 발행 주기가 불규칙하거나 긴 환경에서 변화된 네트워크 상태에 유연하게 대응하며, 네트워크 상태가 불안정한 경우 QoS 수준을 상향 조정함으로써 패킷 손실을 효과적으로 줄이는 것으로 나타났다. 향후 연구에서는 신뢰성과 지연 시간의 상충 관계(trade-off)를 고려하여, 구독자의 요구 특성에 따라 QoS를 더욱 능동적으로 제어하는 방안을 설계하고 실험적으로 검증할 예정이다.

주제어 : MQTT; 브로커 기반 제어, 동적 QoS 제어; 실시간 상태 평가, 적응형 메시지 전달

Abstract In the Internet of Things(IoT) environment, Message Queuing Telemetry Transport(MQTT) is a widely used message transmission protocol based on its lightweight and efficiency. However, in the standard MQTT architecture, subscribers statically configure their Quality of Service(QoS) levels without considering network conditions, which can lead to unnecessary overhead or data loss under depending on changes in transmission environments. In this paper, we propose a new QoS-optimized transmission method where the broker evaluates network conditions in real-time, then dynamically selects and adjusts to the lowest Quality of Service (QoS) level that satisfies the allowable latency and loss rate required by the subscriber's application class before delivering the message. The proposed approach allows the broker to recommend an appropriate QoS level at the initial subscription stage to prevent misconfiguration and adaptively adjust the QoS level at the time of message delivery, thereby improving both efficiency and reliability. Equation-based performance analysis demonstrates that the proposed method adapts flexibly to changing network conditions, especially in environments with irregular or long publishing intervals, and effectively reduces packet loss by increasing the QoS level when the network becomes unstable. Future work will focus on designing and experimentally verifying a method to more actively control QoS according to subscriber requirements by considering the trade-off between reliability and latency.

Key Words : MQTT, broker-based control, dynamic QoS control, real-time condition evaluation, adaptive message delivery

*교신저자 : 경연웅(ywkyung@seoultech.ac.kr)

접수일 2025년 07월 15일 수정일 2025년 08월 01일 심사완료일 2025년 08월 08일

1. 서론

사물인터넷(Internet of Things; IoT)은 헬스케어, 스마트 홈, 스마트 팜 등 다양한 분야에서 핵심적인 기술로 자리잡고 있다[1-3]. IoT 서비스가 급속히 발전함에 따라, 경량성과 에너지 효율성을 특징으로 하는 메시지 기반 프로토콜인 Message Queuing Telemetry Transport(MQTT)를 활용한 다양한 응용 시스템 및 모델이 활발히 개발되고 있다. MQTT는 높은 효율성과 유연성을 바탕으로 IoT 환경에서 사실상 표준 프로토콜로 채택되고 있으며, 다양한 플랫폼과 애플리케이션에 폭넓게 적용되고 있다. 실제로 Amazon Web Services, Google Cloud Platform 등 주요 상용 클라우드 서비스에서도 MQTT를 기반으로 하는 IoT 메시지 통신 기능을 제공하고 있다[4].

MQTT는 발행자(Publisher)와 구독자(Subscriber)로 구성된 Publish/Subscribe 아키텍처를 기반으로 동작하며, 브로커(broker)는 양측 간의 간접적인 메시지 전달을 중개하는 역할을 수행한다. 이러한 구조는 클라이언트 간 직접적인 연결 없이 안정적인 데이터 전송을 가능하게 하며, 확장성을 높이는 데 기여한다[5].

MQTT는 네트워크 환경과 요구사항에 따라 서비스 품질(Quality of Service; QoS) 수준을 선택할 수 있도록 설계되어 있다. 그러나 표준 방식에서는 구독자가 네트워크 상태에 대한 정보를 알지 못한 채 특정 QoS 수준으로 구독하기 때문에, 이로 인해 불필요한 오버헤드나 데이터 손실과 같은 비효율적인 전송이 발생할 수 있다[6].

본 논문에서는 이러한 문제점을 해결하기 위해, 브로커가 실시간으로 네트워크 상태를 평가한 후 해당 상황에 적절한 QoS 수준으로 메시지를 전달하는 동적 QoS 제어 방식을 제안한다. 기존 MQTT 구조에서 구독자가 설정한 QoS는 최대 허용 수준으로만 작용하며, 실제 전달 QoS는 발행자와 구독자의 QoS 중 최솟값으로 자동 결정된다. 제안 방식은 구독자가 토픽을 구독할 때 브로커가 네트워크 상태 정보를 전송함으로써, 구독자가 보다 적절한 QoS 수준을 선택할 수 있도록 유도하며, 메시지 발행 시점에서도 브로커가 네트워크 상태를 평가하여 최적의 QoS 수준으로 메시지를 전달하는 구조를 갖는다. 이를 통해 불필요한 오버헤드와 패킷 손실률을 줄여, 네트워크 품질이 변화하는 환경에서 효과적인 전송 성능을 확보할 수 있음을 수식 기반 성능 분석을 통해 확인하였다.

본 연구의 기대 효과는 다음과 같다.

(1) 구독자의 QoS 선택 오류 방지

구독자가 브로커로부터 네트워크 상태 정보를 제공받아, 보다 합리적인 QoS 수준을 자율적으로 선택할 수 있도록 유도함으로써 초기 설정 오류를 줄일 수 있다.

(2) 브로커의 네트워크 상태 기반 최적화된 QoS 선택 및 전달

브로커가 실시간 네트워크 상태를 반영하여 적절한 QoS 수준을 선택하고 메시지를 전달함으로써, 과도한 오버헤드를 방지할 수 있다.

본 논문은 다음과 같이 구성된다. 2장에서는 QoS 수준에 관련된 연구들을 살펴보고 본 연구의 차별점을 기술하며, 3장과 4장에서는 MQTT QoS 동작 개요와 제안하는 기법을 설명한다. 5장에서는 수식 기반의 성능 평가를 통해 제안 방식과 표준 방식 간의 성능 차이를 분석한다. 6장에서는 결론과 함께 향후 연구 방향을 제시하며 마무리한다.

2. 관련 연구

MQTT는 IoT 환경에서 널리 사용되는 경량 메시지 전송 프로토콜로, 메시지 전송의 신뢰성과 효율성은 QoS 수준에 의해 결정된다. 그러나 현재의 MQTT 구조에서는 QoS 설정이 주로 발행자 또는 구독자에게 일방적으로 맡겨져 있으며, 실시간 네트워크 상태를 반영한 동적 QoS 조정 기능은 부족하다. 이러한 문제를 개선하기 위한 다양한 QoS 최적화 연구들이 기존 MQTT 및 MQTT-SN 환경을 중심으로 수행되어 왔다.

Kim 등[6]은 MQTT의 Retained Message 기능이 신규 구독자에게 오래된 데이터를 전달할 수 있다는 문제를 해결하기 위해 정보 신선도(Age of Information; AoI)를 고려한 유지 메시지 정책인 ARMY를 제안하였다. ARMY는 브로커가 유지 메시지의 AoI를 평가한 후, 일정 기준을 초과할 경우 발행자에게 새로운 메시지를 요청하여 전달하는 방식이며, Q-learning을 기반으로 기존 Retained Message와 비교하였을 때 평균 AoI를 낮추고 신호 오버헤드를 최소화하는 성능을 보였다. 그러나 이 연구는 Retained Message의 효율적 관리에 집중하고 있어, 네트워크 상태에 따른 QoS 조정이나 지연 시간 최적화에 도움을 주지 못한다는 한계가 있다.

Palmese 등[7]은 MQTT-SN 환경에서 클라이언트가 네트워크 상태를 고려하지 않고 QoS를 설정함으로써 비

효율적인 전송이 발생하는 문제점을 개선하기 위해, 네트워크 상태를 평가하고 이를 기반으로 최적의 QoS 수준을 조정하는 QoS 컨트롤러를 제안하였다. 이는 기존의 QoS 설정 방식과 비교하였을 때 패킷 전달률을 개선하고 지연 시간을 최소화하였다. 그러나 이 방식은 단일 발행자가 다중 구독자에게 메시지를 전달하는 경우, 각 구독자의 네트워크 상태가 상이하기 때문에 모든 구독자에게 최적의 메시지 전달을 보장하기 어렵다는 한계가 있다.

Zhang 등[8]은 전력 IoT 환경에서 패킷 손실률과 지연 시간 간의 균형을 고려한 강화학습 기반의 QoS 조정 알고리즘을 제안하였다. 상한 신뢰 경계(Upper Confidence Bound) 알고리즘을 적용하여 네트워크 상태에 따라 최적의 QoS 수준을 동적으로 선택하는 구조로, 기존의 고정 QoS 설정과 비교하였을 때, 패킷 손실률과 지연 시간을 동시에 줄어드는 효과가 나타났다. 그러나 이 연구는 전력 시스템에 특화된 설계를 기반으로 하여, 일반적인 MQTT 서비스에 적용하기 제약이 따른다.

Lee 등[9]은 MQTT 공유 구독(Shared Subscription) 기능에서, 기존 연구들[10, 11]이 사용한 라운드 로빈 방식은 혼잡한 구독자가 대표로 선정될 경우 메시지 손실 및 지연을 일으킨다는 문제를 지적하였다. 그리고 이를 해결하기 위해, 브로커가 구독자들의 상태 정보를 수집하여 최적의 대표 구독자를 동적으로 선정하는 방안을 제안하였다. 이 방식은 대표 구독자 선정 과정에서 구독자의 혼잡 상태를 고려함으로써 메시지 전달의 안정성을 높이는 효과를 보였다. 그러나 이 연구는 최적의 대표 구독자를 선정하는 데에만 초점을 맞추고 있어, 실제 메시지를 전달하는 과정에서의 네트워크 상태 변화나 QoS에 대한 고려는 부족하다는 한계가 있다.

요약하면, 기존 연구들은 QoS 수준의 최적화를 위해 다양한 접근을 시도하였지만, MQTT의 특정 기능을 개선하는 데에 집중하거나 네트워크 상태를 구독자의 관점에서 반영하기보다 발행자 중심의 QoS 설정에 집중하는 경향이 있었다. 이로 인해 대부분의 접근 방식은 구독자가 초기에 설정한 QoS 수준을 수동적으로 유지할 뿐, 브로커가 능동적으로 개입하여 전송 품질을 동적으로 제어하는 데에는 한계가 있었다.

본 연구는 이러한 한계를 보완하기 위해, 브로커가 네트워크 상태를 실시간으로 평가하고 이를 기반으로 발행 메시지의 QoS 수준을 동적으로 조정하여 구독자에게 전달하는 새로운 방식의 QoS 제어 프레임워크를 제안한다. 발행자-구독자 중심의 QoS 수준 설정에서,

브로커 중심의 동적 조정을 통해 네트워크 상태에 최적화된 QoS 수준으로 메시지를 전달하는 것을 목표로 한다.

3. MQTT QoS 동작 개요

MQTT는 클라이언트-서버 기반의 발행/구독 메시징 전송 프로토콜로, 제한된 대역폭과 낮은 지연이 요구되는 IoT 환경에서 널리 사용되고 있다. MQTT 클라이언트는 발행자와 구독자이며, 그 중심에는 브로커가 있다. 구독자는 자신이 수신하고자 하는 토픽과 함께 QoS 수준(0, 1, 2)을 지정하여 브로커에 구독 요청을 보낸다. 발행자는 QoS 수준을 지정하고 메시지를 발행하며, 발행 메시지를 받은 브로커는 발행자와 구독자가 지정한 QoS 중 더 낮은 수준으로 구독자에게 메시지를 전달한다[12].

MQTT의 QoS 수준별 메시지 전송 흐름은 다음과 같다.

- QoS 0(At most once):

발행자는 브로커에게 PUBLISH 패킷을 1회 전송하며, 브로커는 구독자에게 별도의 확인 메시지 없이 PUBLISH 패킷을 전달한다. 전달에 대한 보장이 없기 때문에 이 과정에서 데이터 손실이 발생할 수 있다.

- QoS 1(At least once):

발행자는 브로커에게 PUBLISH 패킷을 전송한 뒤, 브로커로부터 PUBACK 패킷을 수신해야 한다. 동일하게 브로커는 구독자에게 PUBLISH 패킷을 전송한 후, 구독자의 PUBACK 패킷을 수신해야 하기 때문에 최소 한 번 전달될 것을 보장한다.

- QoS 2(Exactly once):

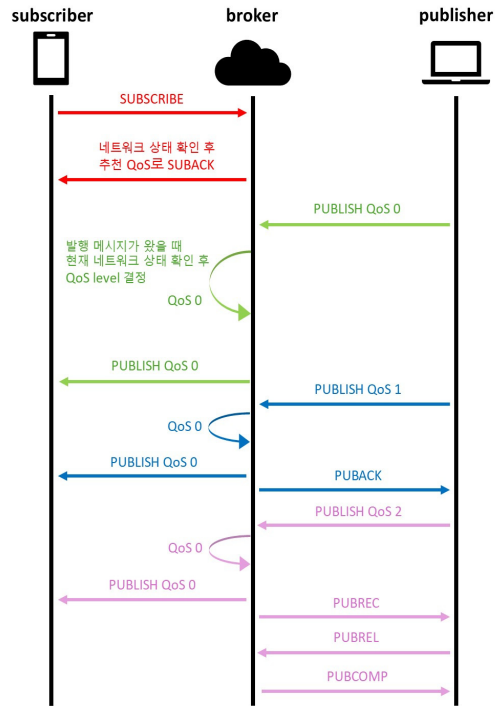
발행자는 PUBLISH 패킷을 전송한 후 PUBREC 패킷을 수신한다. 그 후 PUBREL 패킷을 전송하고 브로커로부터 PUBCOMP 패킷을 수신한다. 구독자 측도 동일한 단계의 메시지 교환을 수행하여 각 메시지가 단 한 번만 전달될 것을 보장한다. 이 과정은 4단계의 제어 메시지를 필요로 하므로 가장 높은 신뢰성을 제공하지만, 상대적으로 높은 지연과 오버헤드가 발생한다.

표준 방식에서 발행 메시지는 설정된 QoS 수준에 따라 위와 같은 절차를 거쳐 전달된다.

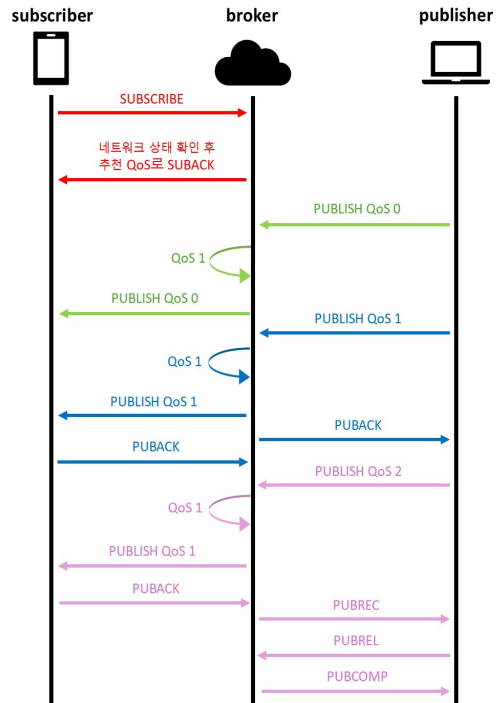
4. 제안 기법

일반적으로 MQTT 연결은 장시간 유지되며, 구독 메시지는 세션 수립 초기에 1회만 전송되기 때문에 표준 방식은 구독자가 구독 시 지정한 QoS 수준으로 계속 구독하고 있다. 따라서 구독자가 설정한 QoS가 현재 네트워크 상태에 적절하지 않을 경우, 불필요한 오버헤드나 데이터 손실이 발생할 수 있다. 특히, 기존 방식은 QoS 0으로 구독한 후 네트워크가 불안정해질 경우, TCP 세션 단절 등으로 인해 발행 메시지가 손실되면 해당 패킷은 복구되지 않고 완전히 소실될 가능성이 있다. 따라서 본 논문에서는 발행자가 메시지를 전송할 때마다 브로커는 네트워크 상태를 실시간으로 평가하여, 각 메시지를 최적의 QoS 수준으로 동적으로 조정해 구독자에게 전달하는 방식을 제안한다. 이를 위해 브로커는 메시지 전송 지연(Transmission Delay)과 패킷 손실률(Packet Loss)을 실시간으로 평가하며[7], 구독자는 사전에 자신의 애플리케이션 클래스가 요구하는 허용 지연(L_{max})과 허용 손실률($P_{loss,max}$)을 서비스 수준 협약(SLA)으로 정의한다. 브로커는 측정된 네트워크 상태를 해당 SLA와 비교하여, 요구사항을 만족하는 가장 낮은 QoS 수준을 동적으로 선택하고 메시지를 전달한다. 이러한 방식은 구독 시점 이후 오랜 시간이 지나 발행 메시지가 도착해도 브로커에 발행 메시지가 도착할 때마다 네트워크 상태를 판단하기 때문에, 발행 간격이 불규칙하거나 장시간 발행이 없어 네트워크 상태가 변동되어도 변화된 네트워크 상태에 능동적으로 대응할 수 있다. 이는 브로커가 네트워크 상태가 악화된 경우, QoS 1 또는 QoS 2를 선택하여 메시지를 큐에 저장하고 재전송할 수 있는 기반을 마련함으로써, 데이터 손실 가능성을 현저히 감소시킨다. 또한 제안하는 방식은 처음 구독 시 브로커가 현재 네트워크 상태 정보를 전송한다. 따라서 브로커가 제공한 네트워크 상태 정보를 기반으로 최적의 QoS 수준을 선택하게 되므로, 초기에 부적절한 QoS 0 구독을 방지할 수 있다.

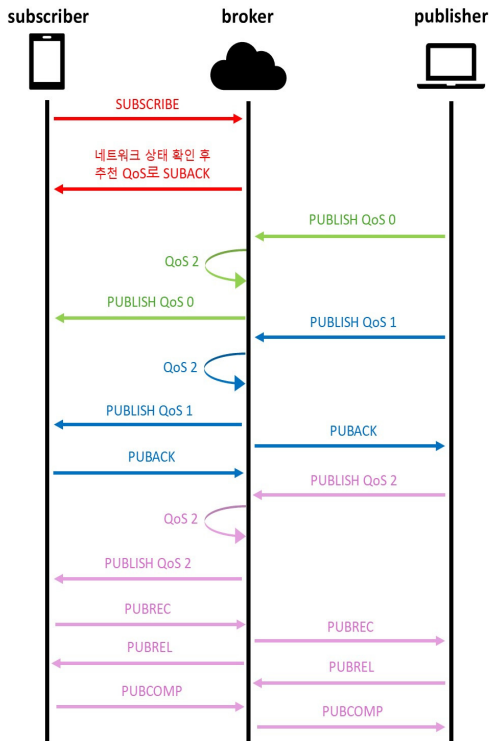
본 논문에서 제안하는 QoS 추천 기반 MQTT 메시지 전송 방식의 동작 흐름은 FIGURE 1, FIGURE 2, FIGURE 3에 나타내었다. 구독자가 브로커에게 SUBSCRIBE 메시지를 전송하면, 브로커는 현재의 네트워크 상태를 평가한 뒤, MQTT 5.0 표준에 따라 SUBACK 메시지의 User Property 필드에 추천 QoS수준을 Key-Value 형식으로 포함하여 구독자에게 전달한다[13]. 구독자는 추천 QoS 수준으로 해당 토픽을 수신하게 된다. 이후 브로



[Fig. 1] The process when the network condition is determined to be QoS 0



[Fig. 2] The process when the network condition is determined to be QoS 1



[Fig. 3] The process when the network condition is determined to be QoS 2

커는 발행자의 메시지를 전달 받을 때마다 수신 시점의 네트워크 상태를 다시 평가하여, 구독자에게 적절한 QoS 수준을 선택하고 메시지를 전달한다. 이러한 방식은 실시간성과 신뢰성이 요구되는 IoT 환경에 적합하다.

5. 성능 분석

제안하는 방식은 구독자의 애플리케이션 클래스가 요구하는 허용 지연 및 허용 손실률을 기반으로 QoS 수준을 동적으로 결정한다. 본 성능 분석에서는 QoS 수준 결정을 위해 Table 1과 같은 의료 애플리케이션 클래스의 요구사항[14]을 가정하였다. 또한 실제 네트워크 환경의 가변성을 반영하고자 메시지 페이로드의 크기를 특정 범위 내에서 임의로 할당하였으며, 페이로드 크기와 QoS 수준이 패킷 손실에 미치는 영향을 고려하여 Table 2와 같이 패킷 손실률을 가정하였다[15]. 마지막으로, 메시지 전송에 소요되는 시간은 다음과 같이 정의하였다. 처리 지연(Processing Delay)은 브로커와 클라이언트가 메시지를 처리하는 데에 소요되는 시간으로 0.1 ms로 설정

하였으며, 전송 지연은 브로커와 클라이언트가 메시지를 전송하는 데에 소요되는 시간으로 Wide-Area 네트워크 환경을 고려하여 7~13 ms의 균등 분포(uniform distribution)를 따르도록 설정하였다[16].

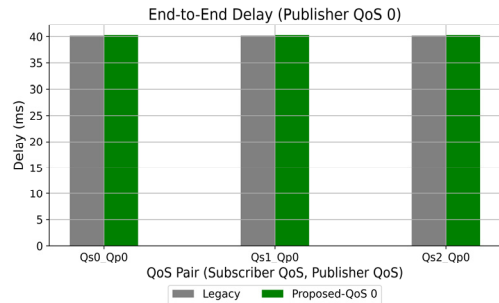
[Table 1] Communication Requirements by Application Class

Data Type	L_{max}	$P_{loss,max}$
2D camera flow	150 ms	10^{-3}
3D camera flow	150 ms	10^{-3}
Audio flow	150 ms	10^{-2}
Temperature	250 ms	10^{-3}
Blood pressure	250 ms	10^{-3}
Heart rate	250 ms	10^{-3}
Respiration rate	250 ms	10^{-3}

[Table 2] Assumed Packet Loss Rate by Payload Size and QoS Level

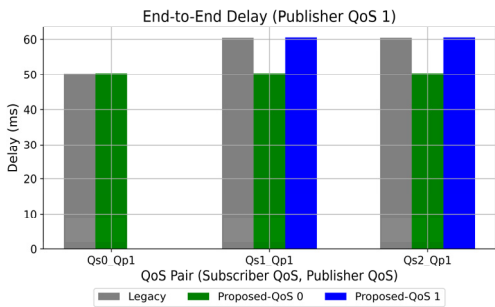
	QoS 0	QoS 1	QoS 2
1000 byte	1.00%	0.90%	0%
2000 byte	1.05%	0.90%	0%
4000 byte	1.20%	0.90%	0%
8000 byte	1.20%	1.05%	0%
16000 byte	1.40%	1.10%	0%

성능 평가로, 구독자와 발행자가 설정하는 QoS 수준의 모든 조합 (Qs, Qp)에 대해 지연과 처리량을 계산하였다. 또한 메시지 재전송을 하는 QoS 1에 대해, 최대 3회까지의 재전송을 가정하고, 각 재전송 시 지연이 누적되게 하여 성능을 평가하였다[17].



[Fig. 4] End-to-end delay comparison based on the subscriber's QoS level or the delivery QoS level determined by the broker, when the publisher sends a message with QoS 0

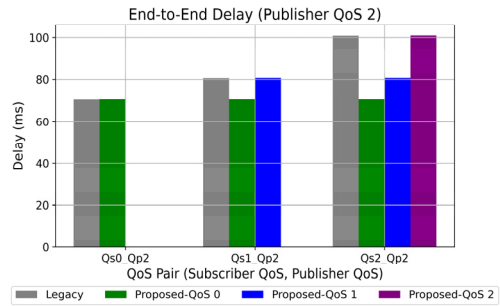
FIGURE 4는 발행자가 QoS 0으로 메시지를 발행하여 표준 방식은 구독자 QoS 수준으로, 제안 방식은 브로커가 판단한 전달 QoS 수준으로 발행 메시지를 전달한 경우의 중단 간 지연을 비교한 결과를 나타낸다. 발행자가 QoS 0으로 발행하였기 때문에 구독자는 어떤 QoS 수준으로 구독을 하였든 발행 메시지를 QoS 0으로 받는다. 따라서 기존 방식은 40.2ms, 제안 방식은 40.3ms의 지연 시간이 걸린다. 이러한 결과는 제안 방식에서 실시간 QoS 결정 단계가 추가되었기 때문에, 처리 지연인 0.1ms가 더해진 것을 나타내며, QoS 0 발행 메시지에 대해서는 구조적으로 개선 효과를 내기 어렵다는 한계를 보인다.



[Fig. 5] End-to-end delay comparison based on the subscriber's QoS level or the delivery QoS level determined by the broker, when the publisher sends a message with QoS 1

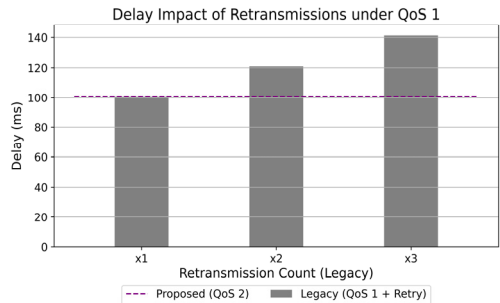
FIGURE 5는 발행자가 QoS 1로 메시지를 발행한 경우의 중단 간 지연을 비교한 그래프이다. 표준 방식은 $\min(Q_s, Q_p)$ 의 결과로 발행 메시지를 받기 때문에, 구독자가 QoS 1 이상으로 구독한 경우 QoS 1로 메시지를 고정적으로 수신하게 된다. 반면, 제안 방식은 브로커가 발행 메시지를 받았을 때 네트워크 상태를 실시간으로 평가하여 QoS 0 또는 QoS 1 중 적절한 수준을 동적으로 선택하여 구독자에게 전달한다. 이러한 방식은 통신 환경이 양호할 경우에는 QoS 0으로 전달하여 전송 지연을 최소화하고, 효율적인 메시지 전달을 가능하게 한다.

FIGURE 6은 발행자가 QoS 2로 메시지를 발행한 경우의 중단 간 지연을 비교한 그래프이다. 실험 결과 Qs2_Qp2 조합에서 기존 방식은 구독자가 구독한 수준인 QoS 2로 발행 메시지를 전달받아 100.8ms의 지연 시간이 걸렸으나, 제안 방식에서는 QoS 0으로 받았을 경우 70.6ms의 지연 시간이 걸리는 것을 확인할 수 있다.



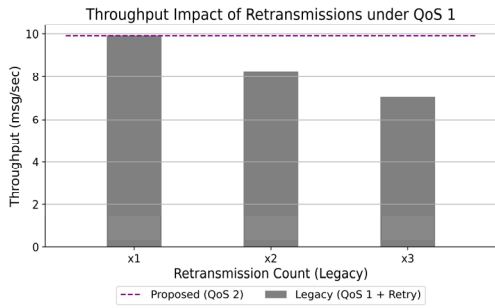
[Fig. 6] End-to-end delay comparison based on the subscriber's QoS level or the delivery QoS level determined by the broker, when the publisher sends a message with QoS 2

이는 네트워크 상태가 양호한 경우 브로커가 QoS 0으로 발행 메시지를 전달하여, 손실 위험이 적을 때 불필요한 오버헤드를 줄이기 때문에, 발행 메시지가 QoS 2인 시나리오에서는 상태 적응적 선택이 더욱 효율적일 수 있음을 보여 준다.



[Fig. 7] End-to-end delay comparison based on the number of retransmissions in a QoS 1 environment

FIGURE 7은 구독자가 QoS 1로 발행 메시지를 전달 받을 때, 재전송 횟수에 따른 중단 간 지연 변화를 나타낸다. 기존 방식은 패킷을 손실할 경우 브로커는 구독자에게 발행 메시지를 재전송하므로, 재전송 횟수가 증가할수록 PUBLISH 및 PUBACK 메시지가 반복되어 지연이 선형적으로 증가한다. 반면, 제안 방식은 브로커가 네트워크 상태를 평가하고 이에 따라 QoS 수준을 상향 조정함으로써, 동일한 환경에서도 메시지를 1회만에 안정적으로 전달 받을 수 있어 지연 증가를 효과적으로 억제할 수 있음을 보여 준다.



[Fig. 8] End-to-end throughput comparison based on the number of retransmissions in a QoS 1 environment

FIGURE 8은 구독자가 QoS 1로 발행 메시지를 전달 받을 때, 재전송 횟수에 따른 중단 간 처리량을 나타내는 그래프로, 단위 시간당 성공적으로 전달한 메시지 수 (msg/sec)로 정의되며, FIGURE 7의 중단 간 지연을 바탕으로 계산되었다. 이 결과는 QoS 수준을 상향 조정함으로써 지연 증가를 일정 수준 감수하더라도, 재전송을 줄이는 것이 전체 처리량 측면에서 더 나은 성능을 달성할 수 있다는 것을 보여 준다. 이로써 제안 방식이 QoS 1 환경에서 메시지 재전송으로 인한 병목 현상을 방지하고, 전체 시스템 처리 효율을 개선하는 데 효과적임을 입증한다.

6. 결론 및 향후 연구

본 논문에서는 구독자가 설정한 QoS 수준이 구독 종료 시까지 고정되어 발행 메시지를 수신하게 되는 표준 MQTT 방식의 한계를 극복하기 위해, 브로커가 네트워크 상태 지표인 메시지 전송 지연과 패킷 손실률을 기반으로 네트워크 상태를 실시간으로 평가하고, 그 결과에 따라 구독자에게 적절한 QoS 수준을 동적으로 선택하여 메시지를 전달하는 새로운 QoS 최적화 메시지 전달 방식을 제안하였다. 제안 방식은 구독자가 특정 토픽을 구독할 때, 브로커가 현재의 네트워크 상태를 평가한 뒤 MQTT 5.0 표준에 따라 SUBACK 메시지의 User Property 필드에 추천 QoS 수준을 Key-Value 형식으로 포함하여 전달함으로써 부적절한 QoS 선택을 예방한다. 또한 발행 메시지를 수신할 때에도 네트워크 상태를 반영하여 최적의 QoS 수준으로 발행 메시지를 전달함으로써 불필요한 오버헤드와 메시지 손실 가능성을 최소화한다. 지연 및 처리율에 대한 수식 기반 성능 분석 결과,

제안 방식은 네트워크 상태가 양호한 상황에서는 QoS 수준을 하향 조정하여 지연 시간을 단축시켰으며, 네트워크 상태가 저하된 상황에서는 QoS 수준을 상향 조정하여 패킷 손실률을 억제하여 안정적인 메시지 전달을 가능하게 하였다. 이러한 결과는 제안 방식이 보다 낮은 중단 간 지연과 높은 처리율을 달성함으로써 효율적인 메시지 전송을 보여 준다. 이처럼 제안 방식은 발행 간격이 긴 IoT 환경이나 네트워크 품질이 변동하는 상황에서 특히 효과적으로 작용할 수 있다. 다만, 예측할 수 없는 통신 장애가 발생할 경우 브로커가 구독자의 네트워크 상태를 실시간으로 정확히 파악하지 못할 가능성이 있다. 이 경우 메시지 손실이 발생하여, 신뢰성을 중시하는 구독자에게는 큰 피해를 초래할 수 있다. 이러한 신뢰성은 중단 간 지연의 최소화과 본질적으로 상충 관계(trade-off) 관계에 있으므로, 향후 연구에서는 신뢰성 중시형·지연 최소화형과 같은 구독자의 요구 특성에 따라 QoS를 능동적으로 조절할 수 있는 방안을 설계하고, 이를 다양한 네트워크 조건에서 실험적으로 검증할 예정이다.

REFERENCES

- [1] P. H. C. Priya, S. G. R. Thenmozhi and C. Srinivasan, "Smart IoT Solutions for Personalized Health: MQTT-Based Blood Pressure Monitoring System," *2023 7th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, Coimbatore, India, pp. 1369-1373, 2023.
- [2] Md. Rokonzaman *et al.*, "Levenberg-Marquardt algorithm-based solar PV energy integrated internet of home energy management system," *Applied Energy*, vol. 378, p. 124407, 2025.
- [3] A. Morchid, R. Jebabra, H. Qjidaa, R. El Alami, and M. O. Jamil, "Agri-tech innovations for sustainability: A fire detection system based on MQTT broker and IoT to improve environmental risk management," *Results in Engineering*, vol. 24, p. 103683, 2024.
- [4] D. -H. Kang *et al.*, "Room Temperature Control and Fire Alarm/Suppression IoT Service Using MQTT on AWS," *2017 International Conference on Platform Technology and Service (PlatCon)*, Busan, Korea (South), pp. 1-5, 2017.
- [5] Soni, Dipa, and Ashwin Makwana, "A survey on mqtt: a protocol of internet of things (iot)," *International conference on telecommunication, power analysis and computing techniques (ICTPACT-2017)*. Vol. 20, 2017.
- [6] Y. Kim and Y. Kyung, "AoI-Aware Retained Message Policy in MQTT-Based IoT Networks," in *IEEE Sensors*

Journal, vol. 24, no. 21, pp. 35809-35819, 1 Nov.1, 2024

- [7] F. Palmese, A. E. C. Redondi, and M. Cesana, Adaptive quality of service control for MQTT-SN, *Sensors*, vol. 22, no. 22, p. 8852, Nov. 2022.
- [8] H. Zhang, H. Zhang, Z. Wang, *et al.*, Delay-reliability-aware protocol adaption and quality of service guarantee for message queuing telemetry transport-empowered electric Internet of Things, *International Journal of Distributed Sensor Networks*, vol. 18, no. 5, 2022.
- [9] S. Lee, Y. Kim, Y. Kim, J. Choi, and Y. Kyung, "Cost-aware Optimal Transmission Scheme for Shared Subscription in MQTT-based IoT Networks," in *Journal of The Korea Internet of Things Society*, vol. 10, no. 4, pp. 1-8, 2024.
- [10] H. Mohammed, Z. Chen and W. Chen, "Energy-Efficient Joint Pushing and Caching Based on Markov Decision Process," in *IEEE Transactions on Green Communications and Networking*, vol.3, no.2, pp.433-445, 2019.
- [11] T. Song and Y. Kyung, "Deep Reinforcement Learning Based AoI-aware Active Queue Management for IoT Sensor Networks," in *IEEE Internet of Things Journal*, vol. 11, no. 9, pp. 16700-16709, May 2024.
- [12] HiveMQ, "MQTT Essentials & MQTT 5 Essentials," 2020.
- [13] OASIS, *MQTT Version 5.0. OASIS Standard*, 2019. [Online]. Available: <https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html>.
- [14] H. N. Qureshi, M. Manalastas, A. Ijaz, A. Imran, Y. Liu, and M. O. Al Kalaa, "Communication requirements in 5G-enabled healthcare applications: review and considerations," *Healthcare*, vol. 10, no. 2, p. 293, Feb. 2022.
- [15] S. Lee et al., "Correlation analysis of MQTT loss and delay according to QoS level," in *Proc. Int. Conf. Inf. Neww. (ICOIN)*, 2013.
- [16] Y. Kyung, H. Ko, J. Lee, S. Pack, N. Park and N. Ko, "Location-Aware B5G LAN-Type Services: Architecture, Use Case, and Challenges," in *IEEE Communications Magazine*, vol. 62, no. 1, pp. 88-94, January 2024.
- [17] F. Al-Turjman and S. Alturjman, "A novel smart healthcare design based on fog and cloud computing paradigms," in *Sensors*, vol. 22, no. 22, p. 8852, Nov. 2022.

정 주 은(Jueun Jeong)

[준회원]



- 2022년 3월 : 공주대학교 정보통신공학과 입학
- 2025년 3월 ~ 현재 : 공주대학교 정보통신공학과 재학

<관심분야>

6G 이동통신, 최적화

박 서 연(Seoyeon Park)

[준회원]



- 2023년 3월 : 공주대학교 정보통신공학과 입학
- 2025년 3월 ~ 현재 : 공주대학교 정보통신공학과 재학

<관심분야>

백엔드, 사물인터넷(IoT), ML

경 연 응(Yeunwoong Kyung)

[중신회원]



- 2011년 2월 : 고려대학교 전기전자공학부(공학사)
- 2016년 8월 : 고려대학교 전기전자공학부(공학박사)
- 2016년 9월 ~ 2020년 3월 : 삼성전자 무선사업부 책임연구원
- 2020년 3월 ~ 2022년 8월 : 한신대학교 컴퓨터공학부 조교수
- 2022년 9월 ~ 2025년 2월 : 국립공주대학교 정보통신공학과 조교수
- 2025년 3월 ~ 현재 : 서울과학기술대학교 전자공학과 조교수

<관심분야>

사물인터넷(IoT), SDN, 5G/6G, 이동성, 모바일 서비스