

# MQTT 기반 IoT 네트워크에서 공유 구독을 위한 비용 관리 최적 전송 방식

이선빈<sup>1</sup>, 김영훈<sup>1</sup>, 김용은<sup>1</sup>, 최재윤<sup>1</sup>, 경연웅<sup>2\*</sup>  
<sup>1</sup>공주대학교 정보통신공학과 학생, <sup>2</sup>공주대학교 정보통신공학과 교수

## Cost-aware Optimal Transmission Scheme for Shared Subscription in MQTT-based IoT Networks

Seonbin Lee<sup>1</sup>, Younghoon Kim<sup>1</sup>, Youngeun Kim<sup>1</sup>, Jaeyoon Choi<sup>1</sup>, Yeunwoong Kyung<sup>2\*</sup>

<sup>1</sup>Student, Division of Information & Communication Engineering, Kongju National University

<sup>2</sup>Professor, Division of Information & Communication Engineering, Kongju National University

**요약** 기술이 발전함에 따라 Internet of Things(IoT)기술 또한 빠르게 발전하고 있다. IoT 기술에는 Message Queuing Telemetry Transport(MQTT)를 포함한 다양한 프로토콜이 사용되고 있다. MQTT는 경량 메시지 프로토콜로 제한된 대역폭과 전력을 가진 환경에서도 효율적으로 데이터를 전송할 수 있어 IoT 분야에서 de-facto 표준 프로토콜로 고려되고 있다. 본 논문에서는 MQTT 5.0의 기능인 공유구독에서 메시지 전송 방식을 개선한 방식을 제안하고자 한다. 공유구독에서 메시지 전송 방식 중 대중적으로 사용되는 라운드 로빈 방식은 Client의 현재 상태를 고려하지 않는다는 단점이 있다. 이러한 단점을 보완하기 위해 본 논문에서는 현재 상태를 고려하여 최적의 전송 방식을 선정하는 방법을 제안한다. 이때 Markov decision process(MDP)를 기반으로 모델링을 수행하고, Q-Learning을 이용하여 최적의 전송 방식을 선정하였다. 시뮬레이션 결과를 통해 제안한 방식과 기존에 사용되는 방식들을 비교하여 다양한 환경에서 성능 분석을 진행해 제안한 방식이 기존 방식들보다 더 좋은 성능을 보이는 것을 확인하고, 향후 연구의 방향성을 제시하면서 본 논문을 마무리하고자 한다.

**주제어** : IoT, MQTT, QL(Q-Learning), 공유 구독, 마르코프 결정 과정

**Abstract** As technology advances, Internet of Things (IoT) technology is rapidly evolving as well. Various protocols, including Message Queuing Telemetry Transport (MQTT), are being used in IoT technology. MQTT, a lightweight messaging protocol, is considered a de-facto standard in the IoT field due to its efficiency in transmitting data even in environments with limited bandwidth and power. In this paper, we propose a method to improve the message transmission method in MQTT 5.0, specifically focusing on the shared subscription feature. The widely used round-robin method in shared subscriptions has the drawback of not considering the current state of the clients. To address this limitation, we propose a method to select the optimal transmission method by considering the current state. We model this problem based on Markov decision process (MDP) and utilize Q-Learning to select the optimal transmission method. Through simulation results, we compare our proposed method with existing methods in various environments and conduct performance analysis. We confirm that our proposed method outperforms existing methods in terms of performance and conclude by suggesting future research directions.

**Key Words** : IoT, MQTT, QL(Q-Learning), Shared Subscription, Markov decision process

This work was supported by the research grant of Kongju National University Industry-University cooperation foundation in 2024.

\*교신저자 : 경연웅(ywkyung@kongju.ac.kr)

접수일 2024년 06월 27일 수정일 2024년 07월 18일 심사완료일 2024년 08월 07일

## 1. 서론

기술이 발전함에 따라 사람들은 효율성을 추구하고 있다. 그 중 주목받는 기술 중 하나는 수동 작업을 최소화하여 무선 네트워크에서 데이터를 수신하고 전송하는 사물인터넷(Internet of Things; IoT)이다. 이 기술은 산업, 물류 및 운수, 농업 등 다양한 분야에서 사용되고 있는 추세이다[1-4].

IoT의 대표적인 예시로는 Smart Home이 있다. Smart Home은 내부 기기들로부터 발생하는 에너지 소비 데이터를 분석하여 에너지 소비에 대한 전력 요금 절감 계획을 제안할 수 있다. 또한 실시간 IoT 응용 프로그램을 통하여 지속적으로 데이터를 분석하고, 고장 기계를 신속하게 감지하여 교체할 수 있다. 이를 통해 Smart Home은 효율적으로 에너지를 관리하고, 기기 유지보수에 도움을 준다[5].

IoT는 일반적으로 제한된 대역폭을 기반으로 데이터를 송수신하며 신뢰성 및 배터리 효율성 등의 문제가 존재하기 때문에 이러한 특성을 고려하여 MQTT, CoAP 등 다양한 IoT 프로토콜들이 소개되었다[6-8].

그 중에서 MQTT 프로토콜은 Client가 Server의 데이터를 요청하면 Server가 처리한 뒤에 응답을 전송하는 방식으로 동작하는 전통적인 Server/Client 방식과 달리 Publish/Subscribe 방식으로 동작한다. Publish/Subscribe 방식은 Client와 Broker로 구성되어 있다. Client는 특정 주제에 대한 정보를 제공하는 Publish와 원하는 주제에 대한 정보를 받아오는 Subscribe를 할 수 있다. 이때 Publish를 하는 Client를 Publisher, Subscribe를 하는 Client를 Subscriber라고 한다. Client는 Publisher와 Subscriber를 둘 다 수행하는 것도 가능해, 특정 주제에 대한 정보를 제공하면서 원하는 주제에 대한 정보를 받아 오는 작업도 수행 가능하다. Broker는 Publisher와 Subscriber간 중개자 역할을 하여 통신을 돕는 역할을 한다. 즉 Publish/Subscribe 통신은 Broker 라는 요소를 통하여 Publisher와 Subscriber간의 통신을 처리하는 방식이다[9-11].

2019년 출시된 MQTT 5.0 버전의 추가된 기능 중 하나인 공유 구독(Shared Subscription)은 여러 Client가 동일한 주제에 대해 메시지를 공유하는 그룹을 만들고, 그룹 내 Client 중 대표 Client가 Broker에게 메시지를 받아 그룹 내 다른 Client들에게 메시지를 전송하는 기능이다[9].

이 때, 공유 구독 시 Broker의 전송 방식은 표준에는

정의되어 있지 않지만 기존 연구들은 라운드 로빈 방식을 채택하여 연구를 진행하고 있다[12].

라운드 로빈 방식은 부하와 상관없이 순차적으로 대표 Client를 선정하여 메시지를 전송하는 방식이다. 하지만 이러한 방식은 Client의 현재 상태를 고려하지 않는다는 단점이 있다. 예를 들어, 현재 혼잡한 Client가 대표 Client로 선정되는 경우에는 Broker로부터 전송받은 메시지를 수신하지 못하거나, 공유 구독 내의 다른 Client들에게 메시지를 전송하는데 시간이 지연되는 경우가 발생할 수 있다.

본 연구에서는 이러한 문제를 해결하기 위해 MQTT 5.0의 기능을 활용하여 Broker가 각 Client들의 현재 상태에 대한 정보를 수집하고, 이 정보를 기반으로 최적의 전송방식을 선정하는 방식을 제안하고자 한다.

본 논문은 다음과 같이 구성되어 있다. 2장에서 시스템 모델과 모델링, 최적의 전송방식을 선정할 때 사용되는 Q-Learning Algorithm을 설명하고, 3장에서 성능평가, 4장에서 결론 및 향후 진행될 연구를 설명하며 마무리된다.

## 2. 본론

### 2.1 시스템 모델

본 논문에서는 Broker가 각 Client들의 현재 수신할 수 있는 메시지의 크기 값인 Receive Maximum Size를 수집하여 Receive Table을 생성하고, 다른 Client들간의 거리 Table을 생성한다. 생성된 두 개의 Table을 기반으로 최적의 Client를 결정하고자 한다. 구체적으로 MQTT 5.0 기능 중 Publish Message에 대한 응답을 위해 사용되는 Response Topic의 헤더 내 사용자 설정 필드를 수정하여 브로커가 Publish Message에 대한 Response Topic Message를 수신할 때 Receive Maximum Size를 함께 수집할 수 있다. 하지만 Table 관리를 통해 최적의 Client를 결정하는 방법은 라운드 로빈 방식에 대비하여 Table 관리의 오버헤드가 발생할 수 있으며, Client들의 빠른 업데이트가 필요한 경우라면 Broker의 부하가 발생하더라도 공유 구독을 하지 않는 것이 더 효율적일 수 있다. 그러므로 본 논문에서는 지연 요구사항 및 전송/컴퓨팅 오버헤드를 고려한 최적의 전송 방법을 결정하고자 한다. 이를 위해 Markov Decision Process(MDP)를 기반으로 모델링을 수행하였고, Q-Learning Algorithm을 통하여 최적의 정책을 결정하고자 한다.

## 2.2 모델화

2.2에서는 2.1에서 설명한 방법을 MDP로 모델링한다.

### A. State Space

상태 집합  $S$ 는 다음과 같이 정의한다.

$$S = R \times T \times M \times D \quad (1)$$

$R = \{2, \dots, N\}$ ,  $r \in R$ 은 Broker가 공유구독 그룹 내 마지막으로 메시지를 전송한 Client를 정의한다. 여기서  $N$ 은 공유 그룹 내의 Client의 수를 정의한다.

$T = \{0, 1, \dots, T_m\}$ ,  $t \in T$ 는 Receive Table의 마지막 업데이트로부터 지난 시간을 정의한다. 만약  $t = T_m$ 이면 Receive Table의 신뢰성이 낮아졌다고 판단하여, Broker가 Client들에게 Connect 패킷을 요청하여 Receive Table을 업데이트 한다. Connect 패킷은 Client가 Broker에게 서버 연결을 요청하는 패킷으로, 이 패킷에 Receive Maximum Size를 포함시켜 Broker에게 전송한다[3].

$M = \{M_{\min}, \dots, M_{\max}\}$ ,  $m \in M$ 은 메시지의 데이터 크기를 정의한다. 메시지의 데이터 크기  $M$ 은  $M_{\max}$ 를 최대값,  $M_{\min}$ 을 최소값으로 갖는 Uniform Distribution을 따른다[13].

$D = \{0, \dots, D_m\}$ ,  $d \in D$ 는 Broker가 받은 Publish Message의 Delay를 정의한다. Delay는 메시지가 발행된 시점으로부터 경과한 시간으로 정의한다.  $D$ 는 지수 분포를 따른다고 가정한다[7].

### B. Action Space

각 의사 결정 구간에서 Broker는 행동을 결정한다.  $A$ 는 Action Space를 의미하며,  $A = \{0, 1, 2\}$ 로 정의한다.

Action 0은 공유구독 그룹 내의 모든 Client에게 메시지를 전송하는 행동이다.

Action 1은 라운드 로빈 방식을 이용하여 순차적으로 대표 Client를 선정하여 메시지를 전송하는 행동이다.

Action 2는 Broker가 수집한 Receive Table과 거리 테이블을 기반으로 최적의 Client를 선정하여 메시지를 전송하는 행동이다.

### C. Transition Probability

현재 상태  $s$ 에서 Broker가 행동  $a$ 를 선택했을 때 다음 상태  $s'$ 로의 전이확률은 다음과 같이 정의한다.

$$P[s'|s, a] = P[r'|r, a] \times P[t'|t] \times P[m'|m] \times P[d'|d] \quad (2)$$

Broker가 메시지를 전송하면 Client는 메시지를 받는다. 이 때 Broker의 행동과 이전의  $r$ 값을 고려하여  $r'$ 의 값이 설정된다. 따라서  $P[r'|r, a]$ 은 다음과 같이 정의한다.

$$P[r'|a=0] = \begin{cases} 1, & \text{if } r' = 0, \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

$$P[r'|0 < r < N, a=1] = \begin{cases} 1, & \text{if } r' = r+1, \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

$$P[r'|r=N, a=1] = \begin{cases} 1, & \text{if } r' = 1, \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

$$P[r'|a=2] = \begin{cases} 1, & \text{if } r' = r_{a2}, \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

Publisher가 메시지를 발행했을 때 Response Topic이 포함된 메시지가 평균  $\lambda$ 인 포아송 분포를 따른다고 가정한다[14]. 따라서  $\tau$ 시간 동안 Response Topic이 포함된 메시지를 발행할 확률은  $\lambda\tau$ 이다. 따라서  $P[t'|t]$ 는 다음과 같이 정의한다.

$$P[t'|0 \leq t < T_m] = \begin{cases} \lambda\tau, & \text{if } t' = 0, \\ 1 - \lambda\tau, & \text{if } t' = t+1. \end{cases} \quad (7)$$

$$P[t'|t = T_m] = \begin{cases} 1, & \text{if } t' = 0, \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

메시지의 데이터 크기는 Uniform Distribution을 따른다. 따라서  $P[m'|m]$ 은 다음과 같이 정의한다.

$$P[m'|m] = \frac{1}{M_{\max} - M_{\min} + 1} \quad (9)$$

Broker가 받은 Publish Message의 Delay는 지수 분포를 따른다. 따라서  $P[d'|d]$ 는 다음과 같이 정의한다.

$$P[d'|d] = \lambda e^{-\lambda d'} \quad (d' \geq d) \quad (10)$$

### D. Reward

보상 및 비용 함수에 대해 Delay와 전송 Cost, Computing Cost를 고려한다. 보상 및 비용 함수  $r(s, a)$ 는 다음과 같이 정의한다.

$$r(s, a) = \omega g(s, a) - (1 - \omega)f_1(s, a) + f_2(s, a) \quad (11)$$

$f_1(s, a)$ 은 전송 Cost로 Broker가 Client에게 메시지를 전송할 때 사용되는 Cost함수로 다음과 같이 정의한다.  $C_b$ 는 브로커의 전송 Cost,  $C_r$ 은 재전송 Cost,  $C_t$ 는 Response Topic Cost이다.

$$f_1(s, a) = \begin{cases} N \times C_b + C_r + C_t, & \text{if } a = 0, \\ C_b + C_r + \left( \sum_{i=1}^N \text{hop}(i) \times C_c \right) + C_t, & \text{if } a = 1, 2. \end{cases} \quad (12)$$

$f_2(s, a)$ 는 Computing Cost로 Broker가 Table 계산 및 관리를 수행하는 Cost 함수로 다음과 같이 정의한다.  $C_{tm}$ 은 Broker의 Table 계산 Cost이다.

$$f_2(s, a) = \begin{cases} 0, & \text{if } a = 0, 1, \\ C_{tm}, & \text{if } a = 2. \end{cases} \quad (13)$$

$g(s, a)$ 는 Delay를 통한 보상함수로 다음과 같이 정의한다. Delay가 낮을수록 더 높은 보상을 얻게 설정하였다.

$$g(s, a) = - \sum_{x=1}^N (\delta_{cur,x}(t) - \tau_{target})_+ \quad (14)$$

여기서  $\delta_{cur,x}(t)$ 는 현재 시간  $t$ 에서의  $x$ 번째 Client의 메시지 Delay를 나타내고  $\tau_{target}$  서비스 목표 Delay를 나타낸다.  $x_+$ 는 Ramp Function으로 다음과 같이 정의한다.

$$x_+ = \begin{cases} x, & \text{if } x \geq 0, \\ 0, & \text{otherwise.} \end{cases} \quad (15)$$

$\omega$ 는  $g(s, a)$ 와  $f_1(s, a)$ ,  $f_2(s, a)$ 를 균형 있게 조절하는 가중치 계수이다.

### 2.3 Q-Learning Algorithm

2.2에서 정의된 MDP 모델의 최적의 정책을 찾기 위해 본 논문에서는 Q-Learning 기반 알고리즘을 사용하였다. Q-Learning은 강화학습 방법론 중 하나로, 어떠한 상태에서 어떤 행동을 취했을 때 받을 수 있는 예상 보상을 나타내는 값인 Q값을 업데이트 하면서 최적의 행동 정책을 찾아가는 알고리즘이다[15]. Q-Learning에 사용되는 식은 다음과 같다.

$$Q(s, a) \leftarrow Q(s, a) + \alpha [r + \gamma \max_{a'} Q(s', a') - Q(s, a)] \quad (16)$$

$s$ 는 현재 상태,  $a$ 는 Action,  $r$ 은 보상,  $s'$ 은 다음 상태,  $\alpha$ 는 학습률,  $\gamma$ 는 감쇠율을 나타낸다. 탐험과 활용을 조절하는 확률 값  $\epsilon$ 를 사용한다. 이 확률 값에 따라 에이전트는  $\epsilon$  확률로 무작위 행동을 선택하고  $1 - \epsilon$  확률로 Q값이 높은 행동을 선택한다.  $\epsilon$ 는 업데이트를 반복하며 점차 감소하므로 학습 초반에는 탐험을 하고 학습을 할수록 Q값을 활용하게 된다. Q-Learning의 동작과정 및 업데이트 과정은 다음과 같다.

Q-Learning update Algorithm

- 1: 변수 초기화
- 2: 에피소드 반복
- 3: 현재 상태  $s$  확인
- 4: 행동  $a$  선택( $\epsilon$ : 무작위 or  $1 - \epsilon$ : 최적정책)
- 5: 보상  $r$ 과 다음상태  $s'$ 을 얻음
- 6: Q값 업데이트
- 7: 반복 종료
- 8: 업데이트 전의 Q값과 업데이트한 Q값을 비교했을 때 둘의 차이가 특정 값  $\alpha$  보다 작으면  $n = n + 1$
- 9:  $n$ 이 임의로 설정한 반복 횟수가 되면 최적의 정책 계산, 그렇지 않은 경우에는  $\epsilon$  값을 줄이고 다시 2부터 반복

이러한 동작과정 및 업데이트 과정을 통하여 최종적으로 학습한 Q값을 기반으로 각 상태에서 최적의 액션을 선택하는 최적 정책을 찾게 된다.

<Table 1> Simulation Environment

CPU	Intel(R) Core(TM) i7-10750H
GPU	NVIDIA GeForce RTX 2070
RAM	RAM 32GB
OS	Windows 11 Pro
IDE	Visual Studio Code 1.92.1

### 3. 성능평가

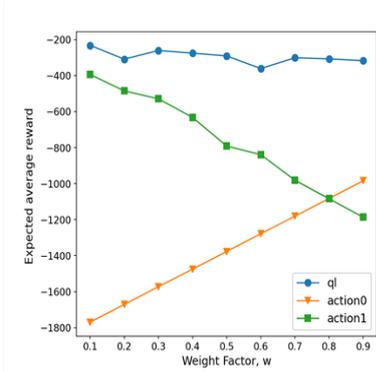
성능을 평가하기 위해, <Table 1>과 같은 환경에서 Python의 Matplotlib를 이용하여 시뮬레이션을 진행하였다. 각 시뮬레이션은 1,000번의 결정을 포함하며, 100개의 시뮬레이션의 평균값을 통해 보상을 평가한다. 우리는 본 논문에서 제안한 방법인 ql을 다음 두 가지 방법과 비교한다.

방법1 (action 0) : 공유구독을 하지 않고 공유구독 Client들에게 직접 메시지를 전송하는 방식인 Action 0으로만 메시지를 전송하는 방법

방법2 (action 1) : 라운드 로빈 방식을 이용한 공유구독 전송 방식인 Action 1으로만 메시지를 전송하는 방법

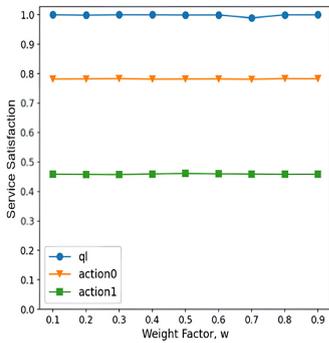
파라미터 값은 다음과 같다. Client 수  $N$ 은 50, Broker의 전송부하  $C_b$ 는 10, Delay Target은 5, 공유구독 내 Client간의 전송 실패확률은 0, Broker가 Client에게 재전송이 일어날 확률 0.1으로 설정하였다.  $Q(s, a)$ 를

업데이트 하기 위해 사용된  $\alpha$ 는 0.2,  $\gamma$ 는 0.95, 에피소드 반복 수는 1000회로 설정하였다. 파라미터 값들의 변화를 주어 다양한 환경에서 각 방법들을 비교하였다.



[Fig. 1] Expected average reward based on Weight Factor,  $w$

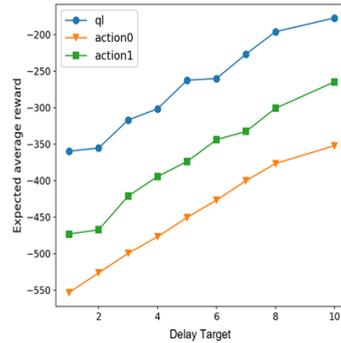
Fig. 1은 가중치  $w$ 에 따른 평균 기대 보상에 대한 그래프이다. action 0 방법은 Delay가 낮지만, Cost가 높기 때문에  $w$ 가 높아질수록 평균 기대 보상이 증가하는 모습을 보인다. action1 방법은 Delay가 높지만, Cost가 낮기 때문에  $w$ 가 높아질수록 평균 기대 보상이 감소하는 모습을 보인다. 우리가 제안한 방법인 ql 방법은 환경에 맞추어 평균 기대 보상이 높은 Action을 선택하고 있으므로, 다른 두 방법보다 높은 평균 기대 보상을 보이고 있다.



[Fig. 2] Service Satisfaction based on Weight Factor,  $w$

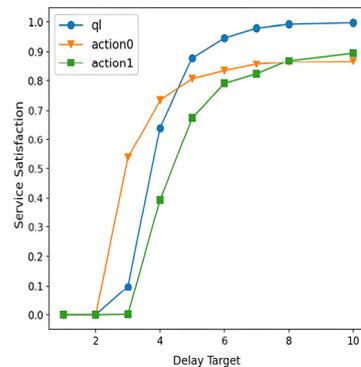
Fig. 2는  $w$ 에 따른 서비스 만족도에 대한 그래프이다. action 0 방법은 Client들에게 직접 메시지를 보내기 때문에 전송에 실패하면 전송 실패한 하나의 Client만 메시지를 못 받는 반면, action 1 방법은 대표 Client에게 전송을 실패할 경우 공유구독 내의 모든 Client가 메시

지를 받지 못하기 때문에 action 0 방법보다 action 1 방법이 더 낮은 서비스 만족도를 보인다. 반면 ql 방법은 최적의 Client에게 메시지를 전송하므로 재전송이 일어날 확률이 낮기 때문에 높은 서비스 만족도를 보이고 있다.  $\omega$ 의 변화는 서비스 만족도에는 영향을 미치지 않기 때문에  $\omega$ 가 변화해도 서비스 만족도는 변화하지 않는 모습을 보인다.



[Fig. 3] Expected average reward based on Delay Target

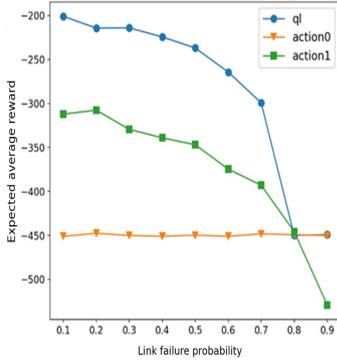
Fig. 3은 Delay Target에 따른 평균 기대 보상에 대한 그래프이다. Delay Target은 목표 Delay값으로, Delay값이 Delay Target값을 초과하거나, 메시지 전송을 실패한 경우에는 서비스를 만족하지 못한 것으로 판단한다. Delay Target값이 증가할수록 서비스 만족도가 증가하므로 모든 방법이 우상향하는 모습을 보인다.



[Fig. 4] Service Satisfaction based on Delay Target

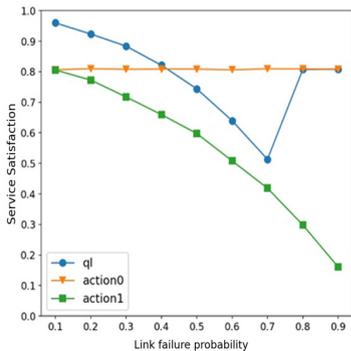
Fig. 4는 Delay Target에 따른 서비스 만족도에 대한 그래프이다. 다른 방법들에 비해 delay가 낮은 action 0 방법은 Delay Target이 2일 때부터 급격하게 만족도가 증가하기 시작하고, delay가 높은 action 1 방법은

Delay Target이 2 ~ 4 사이에서 급격하게 만족도가 증가하고 있다. ql은 action 0 방법과 action 1 방법 사이에서 만족도가 급격하게 증가하고 있다. 이후 만족도는 점차 증가하며, Delay Target이 10일 때 3가지 방법 다 높은 만족도를 보이고 있다.



[Fig. 5] Expected average reward based on Link failure probability

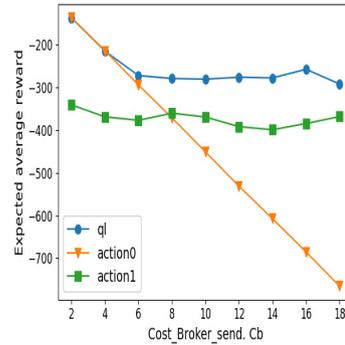
Fig. 5은 공유구독 내 전송 실패 확률에 따른 평균 기대 보상에 대한 그래프이다. 공유구독과 관련이 없는 action 0 방법은 공유구독 내 전송 실패 확률이 증가해도 값이 변하지 않는 모습을 보여준다. 공유구독과 관련 있는 action 1 방법과 ql 방법은 전송 실패 확률이 증가할수록 평균 기대 보상은 감소하는 모습을 보이고 있다. 이 때 ql 방법은 3가지 Action 중 Action 0 의 보상이 가장 커지는 0.8구간 이상부터 Action 0 을 따르는 모습을 보이고 있다.



[Fig. 6] Service Satisfaction based on Link failure probability

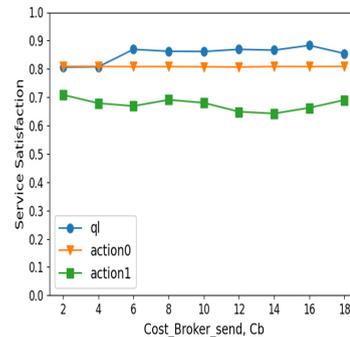
Fig. 6은 공유구독 내 전송 실패 확률에 따른 서비스 만족도에 대한 그래프이다. 공유구독과 관련 없는 action 0 방법은 서비스 만족도 또한 변하지 않는 모습

을 보이고 있다. 공유구독과 관련 있는 action 1 방법과 ql 방법은 전송 실패 확률이 증가할수록 서비스 만족도 또한 감소하는 모습을 보이고 있다. 이 때 ql 방법은 0.8에서 reward가 높은 Action 0 을 선택하므로, 그에 맞춰 만족도 또한 Action 0 의 만족도를 따르는 모습을 보이고 있다.



[Fig. 7] Expected average reward based on Cost\_Broker\_send,  $C_b$

Fig. 7은 Broker 전송 Cost,  $C_b$ 에 따른 평균 기대 보상에 대한 그래프이다. 모든 Client들에게 메시지를 전송해야하는 action 0 방법은  $C_b$ 가 낮을 때는 가장 높은 보상을 보이다가,  $C_b$ 가 증가할수록 보상이 감소하는 모습을 보이고 있다.  $C_b$ 와 큰 관련이 없는 action 1 방법은  $C_b$ 의 변화에도 보상에 큰 변화가 없는 모습을 보이고 있다. ql 방법은  $C_b$ 가 낮을 때에는 3가지 Action중 높은 보상을 취하는 Action인 Action 0을 선택하다가,  $C_b$ 가 증가할수록 보상이 높은 다른 Action을 선택하는 모습을 보인다.



[Fig. 8] Service Satisfaction based on Cost\_Broker\_send,  $C_b$

Fig. 8은 Broker 전송 Cost,  $C_b$ 에 따른 서비스 만족도에 대한 그래프이다. 서비스 만족도 그래프도 마찬가지로  $C_b$ 가 낮을 때에는 ql 방법이 action 0와 같다가  $C_b$ 가 높아질수록 다른 Action을 선택함으로 만족도가 높아지는 모습을 보인다.

#### 4. 결론

본 논문에서는 MQTT 공유구독 내 최적의 전송 방식을 찾아 메시지를 전송하는 방법을 제안한다. 최적의 정책을 찾기 위해 이 논문에서는 문제에 대해 MDP 모델을 정의하고, Q-Learning Algorithm을 이용하였다. 시뮬레이션 결과, 본 논문에서 제안한 방법이 다양한 환경에서 기존에 제안된 방법보다 나은 성능을 보여준다. 향후 실제 IoT 환경을 구현하여 3가지 전송 방식의 데이터 값과 본 연구에서의 결과 데이터 값을 비교하여 검증할 예정이다.

#### REFERENCES

- [1] C.Choi., "A Study On IoT Data Consistency in IoT Environment," *Journal of The Korea Internet of Things Society*, Vol.8, No.5, pp.127-132, 2022.
- [2] Y.Kyung, T.Kim, and Y.Kim, "Retained Message Delivery Scheme utilizing Reinforcement Learning in MQTT-based IoT Networks," *Journal of The Korea Internet of Things Society*, Vol.10, No.2, pp.131-135, 2024.
- [3] H.Ko and Y.Kyung, "Performance Analysis and Optimization of Delayed Offloading System With Opportunistic Fog Node," *IEEE Transactions on Vehicular Technology*, Vol.71, No.9, pp.10203-10208, 2022.
- [4] T.Song and Y.Kyung, "Deep Reinforcement Learning Based AoI-aware Active Queue Management for IoT Sensor Networks," *IEEE Internet of Things Journal*, Vol.11, No.9, pp.16700-16709, 2024.
- [5] H. Hosseinian and H. Damghani, "Smart home energy management, using IoT system," in Proc. 2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI), Tehran, Iran, pp.905-910, 2019.
- [6] S. Kraijak and P. Tuwanut, "A survey on internet of things architecture, protocols, possible applications, security, privacy, real-world implementation and future trends," in Proc. 2015 IEEE 16th International Conference on Communication Technology (ICCT), Hangzhou, China, pp.26-31, 2015.
- [7] Y.Kyung, J.Sung, H.Ko, T.Song, and Y.Kim, "Priority-Aware Actuation Update Scheme in Heterogeneous Industrial Networks," *Sensors*, Vol.24, No.2, pp.1-15, 2024.
- [8] F.Buccafurri, V.D.Angelis, and S.Lazzaro, "MQTT-I: Achieving End-to-End Data Flow Integrity in MQTT," *IEEE Transactions on Dependable and Secure Computing*, Early Access, pp.1-18, 2024.
- [9] OASIS, MQTT Version 5.0. OASIS Standard, 2019.
- [10] K.Govindan and A.P.Azad, "End-to-end service assurance in IoT MQTT-SN," in Proc. 2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC), pp.290-296, 2015.
- [11] E.Longo and A.E.Redondi, "Design and implementation of an advanced MQTT broker for distributed pub/sub scenarios," *Computer Networks*, Vol.224, pp.1-12, 2023.
- [12] H. Mohammed, Z. Chen and W. Chen, "Energy-Efficient Joint Pushing and Caching Based on Markov Decision Process," *IEEE Transactions on Green Communications and Networking*, vol.3, no.2, pp.433-445, 2019.
- [13] A. Antonic and M. Antonic, "Performance Modeling of Publish/Subscribe Middleware Components," in Proc. 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC), Madrid, Spain, pp.1628-1633, 2020.
- [14] L. Hu, Z. Chen, Y. Dong, Y. Jia, L. Liang and M. Wang, "Status Update in IoT Networks: Age-of-Information Violation Probability and Optimal Update Rate," *IEEE Internet of Things Journal*, Vol.8, No.14, pp.11329-11344, 2021.
- [15] M. L. Mari-Altozano, S. S. Mwanje, S. L. Ramirez, M. Toril, H. Sanneck and C. Gijón, "A Service-Centric Q-Learning Algorithm for Mobility Robustness Optimization in LTE," *IEEE Transactions on Network and Service Management*, Vol.18, No.3, pp.3541-3555, 2021.

이 선 빈(Seonbin Lee)

[준회원]



- 2018년 3월 : 공주대학교 정보통신공학과 입학
- 2024년 3월 ~ 현재 : 공주대학교 정보통신공학과 재학

<관심분야>

사물인터넷(IoT), AI, 네트워크

최 재 윤(Jaeyoon Choi)

[준회원]



- 2018년 3월 : 공주대학교 정보통신공학과 입학
- 2024년 3월 ~ 현재 : 공주대학교 정보통신공학과 재학

<관심분야>

ICT융합, 강화학습, 네트워크

김 영 훈(Younghoon Kim)

[준회원]



- 2019년 3월 : 공주대학교 정보통신공학과 입학
- 2024년 3월 ~ 현재 : 공주대학교 정보통신공학과 재학

<관심분야>

ICT융합, 강화학습, 사물인터넷(IoT), AI

경 연 웅(Yeunwoong Kyung)

[종신회원]



- 2011년 2월 : 고려대학교 전기전자전파공학부(공학사)
- 2016년 8월 : 고려대학교 전기전자전파공학부(공학박사)
- 2016년 9월 ~ 2020년 3월 : 삼성전자 무선사업부 책임연구원

■ 2020년 3월 ~ 2022년 8월 : 한신대학교 컴퓨터공학부 교수

■ 2022년 9월 ~ 현재 : 국립공주대학교 정보통신공학과 교수

<관심분야>

사물인터넷(IoT), SDN, 5G/6G, 이동성, 모바일 서비스

김 용 은(Youngun Kim)

[준회원]



- 2019년 3월 : 공주대학교 정보통신공학과 입학
- 2024년 3월 ~ 현재 : 공주대학교 정보통신공학과 재학

<관심분야>

ICT융합, 사물인터넷(IoT), 네트워크