

## 강화학습 기반 Slotted ALOHA의 성능개선

이창규\*, 이승형°

## Performance Improvement of Reinforcement Learning Based Slotted ALOHA

Chang-Kyu Lee\*, Seung-Hyong Rhee°

요약

Framed Slotted ALOHA(FSA)에 강화학습을 적용한 기존의 ALOHA-Q에서는 각 노드가 지능적으로 슬롯을 선택하여 충돌 없이 패킷을 전송한다. 여기서 채널은 여러 개의 프레임으로 구성되고, 프레임은 여러 개의 슬롯으로 구성된다. ALOHA-Q에서는 각 노드가 프레임당 한 개의 슬롯만 선택하기 때문에 프레임 내의 슬롯의 개수, 즉 프레임 크기와 노드 수가 다른 경우 네트워크 성능이 많이 감소한다. 본 논문에서는 기존의 ALOHA-Q에 목적 함수(Objective function) 적용해 각 노드가 프레임 내에서 사용하는 슬롯의 개수를 조절하여 네트워크 성능을 높인다. 시뮬레이션 결과를 통해 논문에서 제안한 방식의 throughput이 노드 수에 민감하지 않고, ALOHA-Q와 비교하여 높은 성능을 내는 것을 보여준다.

**Key Words** : ALOHA protocol, Reinforcement learning, Objective function, ALOHA-Q, Wireless LAN

## ABSTRACT

In the existing ALOHA-Q, which applies reinforcement learning to frame based slotted ALOHA, each node intelligently selects a slot and sends packets without collision. A channel consists of several frames, and a frame consists of several slots. In ALOHA-Q protocol, because each node selects only one slot per frame, network performance is greatly reduced if the frame size and the number of nodes are different. So, in this paper, we apply the objective function to the existing ALOHA-Q to control the number of slots that each node uses within the frame to increase network performance. The simulation results show that the throughput of the proposed algorithm is not sensitive to the number of nodes, and that high performance is produced compared to ALOHA-Q.

## 1. 서론

Additive Links On-line Hawaii Area (ALOHA)는 다중접속을 통한 무선통신을 위해 하와이 대학교에서 개발된 컴퓨터 네트워킹 시스템이다. ALOHA<sup>[1]</sup> 방식은 간단하고 오버헤드가 적다. 하지만 채널에 랜덤하

게 접근하기 때문에 충돌이 많이 발생하게 되고, 충돌에 의한 재전송으로 발생하는 에너지 소모가 크다. 임의접근 방식인 ALOHA 프로토콜에는 Pure ALOHA와 Slotted ALOHA 두 가지 방식이 있다.

Pure ALOHA는 자신이 전송하고자 하는 패킷이 생기면 바로 전송하는 방식이다. 채널이 사용 중인지

\* 본 논문은 한국연구재단의 연구비(2020R1F1A1069399) 지원 및 2018년도 광운대학교 교내 학술연구비 지원에 의해 수행되었습니다.

• First Author : Kwangwoon University, Department of Electronic Convergence Engineering, leefw3006@kw.ac.kr, 학생회원

° Corresponding Author : Kwangwoon University, Department of Electronic Convergence Engineering, rhee@kw.ac.kr, 중신회원  
논문번호 : 202005-108-B-RN, Received May 14, 2020; Revised August 2, 2020; Accepted August 27, 2020

확인하지 않고 패킷을 전송하기 때문에 충돌이 자주 발생하게 된다. 충돌이 발생하게 되면 일정 시간 이후에 전송한다. Pure ALOHA의 throughput을 높이기 위해 나온 Slotted ALOHA는 채널을 여러 개의 타임 슬롯으로 나누고 슬롯의 시작 부분에서 패킷을 전송한다. 슬롯의 시작 부분에서만 패킷을 전송할 수 있어서 Pure ALOHA와 비교해 충돌이 적게 발생하는 장점이 있다. 그러나 Slotted ALOHA도 마찬가지로 채널이 사용 여부를 확인하지 않고 패킷을 전송하기 때문에 다른 프로토콜에 비해 여전히 throughput이 낮다.

ALOHA-Q<sup>[2]</sup>는 FSA에 강화학습을 적용했다. 프레임은 여러 개의 슬롯으로 구성되어 있다. 각 단말은 이전에 전송한 기록을 통해 학습하여 충돌이 발생하지 않는 슬롯을 찾아서 전송한다. 학습 초반에는 랜덤하게 패킷을 전송하지만, 학습이 완료된 후에는 모든 단말이 충돌 없이 패킷 전송이 가능하다. ALOHA-Q는 스케줄링을 위한 추가정보의 교환이 필요하지 않기 때문에 오버헤드가 발생하지 않고 무경쟁 접근 방식처럼 동작할 수 있다. 그러나 ALOHA-Q에서 프레임 크기는 고정되어 있고 프레임당 하나의 패킷을 전송한다. 따라서 노드 수가 네트워크 성능에 크게 영향을 미친다. 노드 수보다 프레임 크기가 큰 경우에는 사용되지 않는 슬롯이 생기게 된다. 반대로 노드 수보다 프레임 크기가 작은 경우 학습이 잘 안 되고 충돌이 발생한다.

본 논문에서는 강화학습<sup>[3]</sup>을 통해 효과적으로 슬롯을 선택하고, 프레임당 보내는 패킷의 개수를 유동적으로 조절해서 네트워크의 성능을 높인다. 시뮬레이션 결과를 통해 각 노드가 일정한 개수의 패킷을 충돌 없이 전송하는 것을 볼 수 있다.

## II. 관련 연구

### 2.1 Dynamic Framed Slotted ALOHA (DFSA)

DFSA는 FSA 알고리즘의 문제점을 해결하기 위해 제안된 알고리즘이다. 기존 FSA 알고리즘은 프레임 크기가 고정되어 있기 때문에 노드 수가 프레임 크기보다 작거나 큰 경우에 성능이 저하된다. 반면에, DFSA 알고리즘은 노드 수를 추정하여 프레임 크기를 유동적으로 변화시킨다. Vogt<sup>[4]</sup> 알고리즘은 이전 프레임에서 충돌이 발생한 슬롯 개수를 이용해서 다음 프레임 크기를 정한다. 충돌이 발생하면 해당 슬롯에 최소 두 개 이상의 노드가 전송한 것을 알 수 있다. 따라서 프레임 크기를 이전 단계 충돌 슬롯 개수에 2를

곱한 값으로 결정한다.

노드 수 추정을 통해 프레임 크기를 결정하는 알고리즘<sup>[5]</sup>에서는 충돌이 발생한 슬롯의 개수와 프레임 크기의 비율을 이용하고 식은 다음과 같다.

$$C_{ratio} = 1 - (1 - \frac{1}{L})^n (1 + \frac{n}{L-1}) \quad (1)$$

L은 이전 프레임 크기이고,  $C_{ratio}$ 는 충돌 비율을 의미하고 n은 추정하고자 하는 노드의 수다. 위와 같이 DFSA 알고리즘은 노드 수 추정을 통해 프레임 크기를 변경해서 성능을 높인다. 그러나 DFSA 알고리즘도 랜덤하게 슬롯을 선택하기 때문에 트래픽이 많은 환경에서 throughput이 현저히 감소한다.

### 2.2 ALOHA-Q

ALOHA-Q는 FSA에 강화학습을 적용하여 각 노드가 지능적으로 슬롯을 선택해 충돌 없이 패킷을 전송하도록 한다. 여기서 채널은 여러 개의 프레임으로 구성되고, 프레임은 여러 개의 슬롯으로 구성된다. 각 노드는 프레임 내 각각의 슬롯에 대해 Q값을 가지고 있다. 노드들은 Q값이 0일 때 임의적으로 시작해서 trial and error를 통해 학습을 진행한다. 전송 결과에 따라 Q값이 업데이트가 되고 각 노드는 전송에 성공할 확률이 높은 슬롯을 찾게 된다. 따라서 별도의 스케줄링 없이 무경쟁 접근 방식처럼 동작한다.  $Q_i^t(a)$ 는 노드 i에서 슬롯 a에 대한 가치를 나타내고, 식은 다음과 같다.

$$Q_i^{t+1}(a) = Q_i^t(a) + \alpha(r - Q_i^t(a)) \quad (2)$$

Q값은 기존의 Q값과 전송 결과(success or failure)로 받게 된 보상(Reward)을 통해 계산된다.  $\alpha$ 는 학습률(learning rate)이고, r은 보상이다. 행동(Action)은 패킷을 전송하고자 하는 슬롯이다. 슬롯 a에서 전송에 성공하면 보상으로 +1을 받고, 충돌이 발생하면 보상으로 -1을 받게 된다. 노드들은 Q값이 높은 슬롯에서 패킷을 전송한다. 그림 1은 전송 결과에 따라 Q값을 업데이트하는 예제이다.  $\alpha = 0.1$ 일 때 프레임 1에서 전송에 성공한 슬롯은 프레임 2에서 Q값이 0.1이 된다. 프레임 2에서 Q값이 가장 높은 슬롯에 전송하고, 충돌이 발생하면 프레임 3에서 해당 슬롯의 Q값은 -0.01로 업데이트가 된다. 따라서 다른 슬롯에

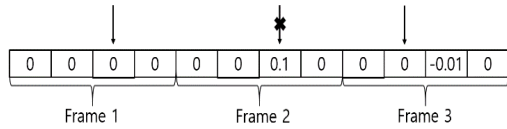


그림 1. Q값 업데이트 예제  
Fig. 1. Example of Q value update

$$Reward = \begin{cases} -1, & ratio > 0 \text{ and } < \frac{1}{4} \\ -2, & ratio \geq \frac{1}{4} \text{ and } < \frac{1}{2} \\ -4, & ratio \geq \frac{1}{2} \text{ and } < \frac{3}{4} \\ -8, & ratio \geq \frac{3}{4} \end{cases} \quad (6)$$

패킷을 전송한다.

### 2.3 기계학습 기반의 ALOHA 프로토콜

최근 여러 논문에서 기계학습 기반의 다양한 ALOHA 프로토콜을 제안하였다. 백오프 기법을 사용하는 Slotted ALOHA에 기계학습을 적용한 연구<sup>6)</sup>에서는 제안한 방식을 통해 최적의 경쟁 윈도우 크기를 찾아서 시스템의 성능을 향상시켰다. 이진 지수 백오프(BEB: Binary Exponential Backoff) 기반의 Slotted ALOHA에서 각 노드는 전송할 패킷이 있으면 경쟁 윈도우 크기 내에서 백오프 슬롯을 선택한 후, 해당 슬롯에 도착하면 패킷을 전송한다. 충돌이 발생하면 경쟁 윈도우 크기가 2배가 되고, 전송에 성공하면 경쟁 윈도우 크기의 최솟값으로 설정된다. 제안하는 기법에서 시간을 여러 개의 프레임으로 나누고 하나의 프레임은 BEB 구간, Broadcast slot과 C-CWS 구간으로 구성된다. BEB 구간에서는 각 노드가 BEB 방식으로 패킷을 전송하고, 이때의 빈 슬롯, 충돌 슬롯과 전송 성공 슬롯의 비율을 파악한다. BEB 구간이 끝나게 되면 미리 학습된 기계학습 모델을 이용해 최적의 경쟁 윈도우 크기를 찾고, 이 값을 C-CWS 구간에서 고정된 경쟁 윈도우 크기로 사용한다.

강화학습 기반의 DFSA 연구<sup>7)</sup>에서는 Q-learning을 통해 프레임 크기를 결정한다. 상태(State)는 이전 프레임에서 충돌 슬롯의 개수이고 행동은 다음과 같다.

$$A1 = 1.46 \times 2.0 \times N_c \quad (3)$$

$$A2 = 1.46 \times 2.2 \times N_c \quad (4)$$

⋮

$$A11 = 1.46 \times 4.0 \times N_c \quad (5)$$

행동의 개수는 총 11개이고  $N_c$ 는 충돌 슬롯의 개수이다. 보상(Reward)은 충돌 비율을 이용해서 구할 수 있다.

### III. 성능개선을 위한 목적함수 적용

기존의 ALOHA-Q는 각 노드가 프레임당 하나의 슬롯에만 전송하기 때문에 프레임 내의 슬롯의 개수, 즉 프레임 크기와 네트워크의 노드 수가 다른 경우 throughput이 감소한다. 노드 수가 프레임 크기보다 작은 경우 패킷을 전송하지 않는 빈 슬롯이 생기고, 큰 경우에는 충돌이 계속 발생하여 학습되지 않는다. 본 논문에서는 프레임 크기를 충분히 크게 하고, 각 노드가 프레임당 보내는 패킷의 개수를 효율적으로 조절하여 성능을 높인다.

#### 3.1 슬롯의 개수 결정

프레임당 사용할 슬롯의 개수(행동의 개수)는 목적 함수를 이용해서 결정한다. 즉, 사용할 슬롯의 개수가 3인 경우 Q값이 높은 3개의 슬롯을 선택해서 패킷을 전송한다. 각 노드는 목적 함수를 가지고 있고 다른 노드와 협력 없이 이 값을 최대화하려고 한다. 목적 함수는 다음과 같다.

$$U_i = S_i \times (C_i - S_i) \quad (7)$$

노드  $i$ 가 한 프레임 내에서 사용하는 슬롯이 많을수록 throughput이 증가하게 된다. 따라서 노드  $i$ 가 프레임당 전송하는 슬롯의 개수인  $S_i$ 를 변수로 사용한다. 하지만 다른 노드가 사용하는 슬롯 개수를 고려하지 않고  $S_i$ 를 증가시키면 충돌 확률 또한 증가하게 된다. 따라서 이전 프레임에서 다른 노드들이 사용하지 않은 슬롯, 즉 빈 슬롯의 개수  $C_i$ 를 이용해 목적 함수를 설계하여  $S_i$ 를 조절한다. 노드  $i$ 가 프레임당 사용할 수 있는 슬롯의 개수는 최대  $C_i$ 이고,  $S_i$ 의 범위는  $1 \leq S_i \leq C_i$ 이다. 각 노드는 carrier sense를 통해 다른 노드가 사용하지 않은 빈 슬롯의 개수를 알 수 있다. 목적 함수  $U_i$ 를 최대화하기 위한 최적의  $S_i$



### IV. 시뮬레이션

제안한 방식의 성능분석을 위해 모든 시뮬레이션은 Python으로 진행하였다. 표 1은 시뮬레이션 파라미터를 나타내고, 모든 노드는 항상 전송할 데이터를 갖고 있다고 가정한다.

그림 3은 노드 수에 따른 ALOHA-Q와 DFSA<sup>[5]</sup>, 그리고 제안한 방식의 normalized throughput을 비교한 그래프이다. Exploration-Exploitation 알고리즘으로는 UCB를 사용하고, 학습률은 0.001로 시뮬레이션을 진행하였다. ALOHA-Q의 프레임 크기는 5인 경우와 25인 경우로 나눠, 각각 프레임 크기가 노드 수보다 작을 때와 클 때의 성능을 비교하였다. 제안한 알고리즘의 프레임 크기는 100이다. ALOHA-Q의 프레임 크기가 5로 노드 수가 같을 때 normalized throughput이 1에 가까운 성능을 보여준다. 하지만 노드 수가 프레임 크기보다 많아지면 성능이 급격하게 떨어지는 것을 확인할 수 있다. ALOHA-Q의 프레임 크기가 25, 노드 수가 20개인 경우 normalized throughput은 약 0.8이다. 노드 수가 작아질수록 낭비되는 슬롯이 많아져서 성능이 감소한다. 그래프를 통해 ALOHA-Q는 프레임 크기와 노드 수가 다른 경우 충돌 또는 슬롯의 낭비로 인해 성능이 감소하는 것을 확인하였다. DFSA 알고리즘은 노드 수를 추정해 프레임 크기를 유동적으로 변화시키지만, 랜덤하게 전송하기 때문에 트래픽이 많은 환경에서 throughput이 상대적으로 낮은 것을 확인할 수 있다.

반면에 제안한 알고리즘은 노드 수가 프레임 크기보다 작더라도 여러 개의 슬롯을 사용하기 때문에 일정 이상의 성능을 유지한다. 먼저, 노드 수가 5인 경우 normalized throughput이 약 0.8, 노드 수가 15인 경우 약 0.93이고 노드 수가 20인 경우 약 0.95로 가장 높은 성능을 보인다. 이는 식 (13)에서 계산한 이론값과 일치하는 것을 알 수 있다. 또한, 노드 수 N이 커질수록 사용하는 슬롯이 많아져서 throughput이 증가하는 것도 시뮬레이션을 통해 확인했다.

그림 4는 그림 3과 같은 조건에서 각 알고리즘의

표 1. 시뮬레이션 파라미터  
Table 1. Simulation parameters

Parameters	Values
Channel bit rate	250 kbits/s
Slot length	1100 bits
Data packet length	1044 bits
ACK packet length	20 bits

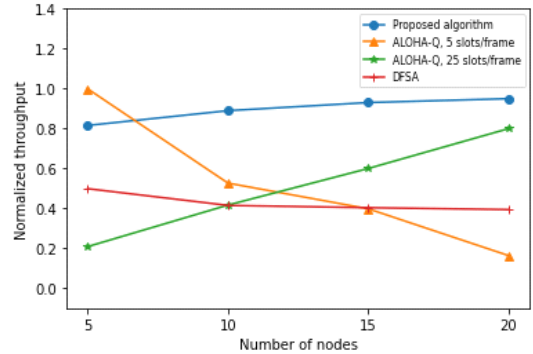


그림 3. Normalized throughput  
Fig. 3. Normalized throughput

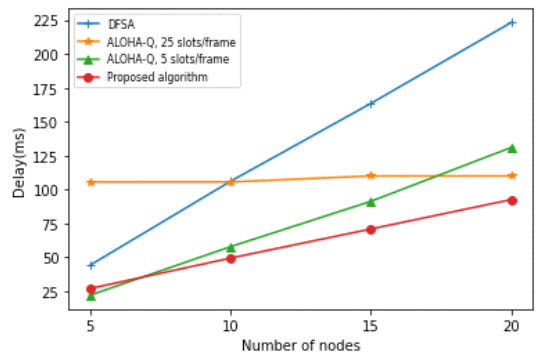


그림 4. 평균 지연 시간  
Fig. 4. Average delay

전송 지연을 측정된 그래프다. 노드의 수가 20개일 때 제안한 알고리즘의 지연 시간은 약 92ms로 가장 짧고 DFSA의 지연 시간은 약 223ms이다. 프레임 크기가 25인 ALOHA-Q는 노드의 수가 5개일 때, 20개의 슬롯이 낭비되기 때문에 전송 지연이 약 105ms로 비교적 길다.

그림 5는 제안한 알고리즘을 사용했을 때 각 노드가 공평하게 슬롯을 사용하는지 확인하기 위한 그래프이다. 각 노드의 평균 throughput을 측정하여 Jain's fairness index<sup>[8]</sup>를 계산했다. fairness index는 0과 1 사이의 값이고, 1에 가까울수록 자원을 공평하게 사용하는 것을 의미한다. 노드의 수가 5개일 때 fairness index는 1이고, 20개일 때 fairness index는 약 0.97이다. 그래프를 통해 제안한 알고리즘을 사용했을 때 모든 노드가 균등하게 슬롯을 사용함을 알 수 있다.

그림 6은 다양한 Exploration-Exploitation 알고리즘에 따라 학습 초반인 time step 0부터 100까지 누적 보상을 나타낸다. 우선 greedy 방식은 Q값이 가장 높은 행동을 선택하여 전송하는 방식이다.

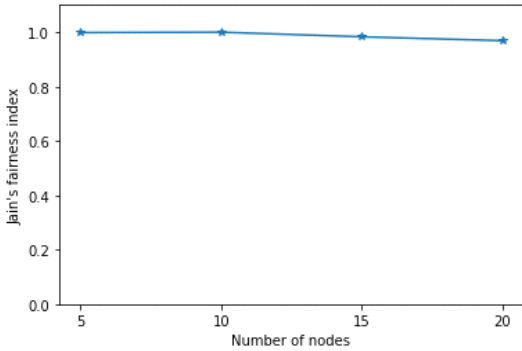


그림 5. Jain's fairness index  
Fig. 5. Jain's fairness index

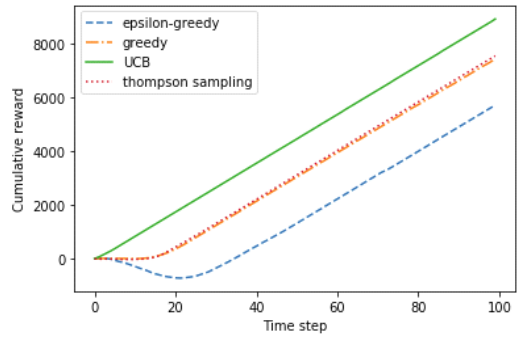


그림 6. 누적 보상  
Fig. 6. Cumulative reward

epsilon-greedy 방식은 greedy 방식과 다르게 1-epsilon의 확률로 Q값이 가장 높은 행동을 선택하고, epsilon의 확률로 랜덤하게 행동을 택한다. Thompson sampling 방식은 확률적으로 행동을 선택한다. 각 행동의 성공 확률을 베타 분포로 표현하고, 샘플링을 통해 샘플 값이 높은 행동을 선택한다. 시뮬레이션 결과를 통해 epsilon-greedy 방식의 time step 100에서의 누적 보상이 약 5500으로 가장 낮은 것을 확인할 수 있다. epsilon-greedy는 epsilon의 확률로 랜덤하게 행동을 선택하기 때문에 Q값이 낮은 행동을 선택해서 누적 보상이 낮을 수 있다. Thompson sampling과 greedy 방식은 누적 보상 증가 폭이 유사하고, time step 100에서의 보상은 약 7000으로 epsilon-greedy 방식보다 1500정도 높다. UCB를 적용해 학습했을 때 time step 100에서의 누적 보상이 약 9000으로 가장 높고 Thompson sampling과 greedy 방식보다 2000, epsilon-greedy 방식보다 3500 정도 높다.

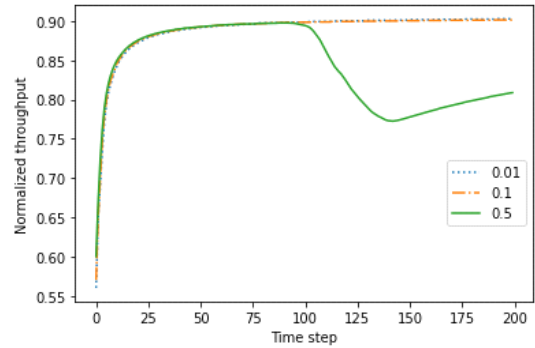


그림 7.  $\alpha$ 에 따른 normalized throughput  
Fig. 7. Normalized throughput according to  $\alpha$

Q값이 높더라도, 한 번이라도 충돌이 발생하면 Q값이 낮아져 다른 슬롯에 패킷을 전송한다. 그 결과 학습률이 높은 경우에는 학습이 잘 안 되고 throughput이 큰 폭으로 감소한다.

## V. 결론

그림 7은 Exploration-Exploitation 알고리즘으로 UCB를 사용하고 프레임 크기가 100, 노드 수가 10일 때 time step 0부터 200까지의 학습률  $\alpha$ 에 따른 normalized throughput을 나타낸다. 학습률이 0.01, 0.1로 작은 경우에 모두 비슷한 성능을 내고, time step이 약 100일 때 normalized throughput이 약 0.9에 수렴하는 것을 확인할 수 있다. 학습률이 0.5인 경우에 time step 100에서 normalized throughput이 약 0.9이고 time step 200에서 약 0.8로 오히려 감소한 것을 볼 수 있다. 학습률이 0.5로 큰 경우에는 과거에 받은 보상의 영향이 적고 현재 time step에서 받은 보상의 영향의 상대적으로 크기 때문에, 현재 time step에서 선택한 행동의 결과에 따라 Q값이 큰 폭으로 변화한다. 따라서 이전 time step에서 성공을 많이 해서

본 논문에서는 기존 ALOHA-Q가 프레임 크기와 노드 수가 다를 때 throughput이 많이 감소하는 문제점을 개선하기 위해 새로운 방식의 Medium Access Control 방식을 제안했다. 기존의 ALOHA-Q에 목적 함수를 적용하여 각 노드가 프레임 내에서 사용하는 슬롯의 개수를 조절했다. 시뮬레이션 결과를 통해 제안한 방식은 노드 수와 프레임 크기가 달라도 throughput이 많이 감소하지 않고, 노드 수가 프레임 크기에 가까워질수록 throughput이 증가하는 것을 확인할 수 있다. 제안한 알고리즘의 fairness index를 측정하여 각 노드가 자원을 공평하게 사용하는 것을 확인하였고, 다양한 Exploration-Exploitation 알고리즘을 적용하여 성능을 비교했다. 또한 학습률에 따른

throughput을 측정하여 비교했다.

향후에 cooperative reinforcement learning을 적용한다면 학습 시간을 더욱 줄이고, 노드 수가 적어도 빈 슬롯 없이 패킷을 전송하여 시스템의 성능을 높이는 것이 가능할 것이다.

## References

- [1] N. Abramson, "The aloha system: another alternative for computer communications," in *Proc. AFIPS '70 (Fall)*, pp. 281-285, New York, United States, Nov. 1970.
- [2] Y. Chu, et al., "Application of reinforcement learning to medium access control for wireless sensor networks," *Eng. Appl. Artif. Intell.*, vol. 46, pp. 23-32, Nov. 2015.
- [3] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, MIT Press, 1998.
- [4] H. Vogt, "Efficient object identification with passive RFID tags," in *Proc. Int. Conf. Pervasive Computing*, pp. 98-113, 2002.
- [5] J. R. Cha and J. H. Kim, "ALOHA-type anti-collision algorithm using tag estimation method in RFID system," *J. KICS*, vol. 30, no. 9, pp. 814-821, Sep. 2005.
- [6] H. J. Kwon, J. H. Lee, and D. G. Jeong, "Machine learning based backoff scheme for slotted- ALOHA," *J. KICS*, vol. 45, no. 1, pp. 34-37, Jan. 2020.
- [7] M. Loganathan, et al., "Energy efficient anti-collision algorithm for the RFID networks," *Bulletin of Electrical Eng. and Informatics*, vol. 8, no. 2, pp. 622-629, Jun. 2019.
- [8] R. Jain, D. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer system," DEC-TR-301 Tech. Rep., Sep. 1984.

## 이 창 규 (Chang-Kyu Lee)



2019년 : 광운대학교 공학사  
2019년~현재 : 광운대학교 대학원 석사과정  
<관심분야> 무선 네트워크 프로토콜, 인공지능/강화학습  
[ORCID:0000-0001-6016-112X]

## 이 승 형 (Seung Hyong Rhee)



1988년 : 연세대학교 공학사  
1990년 : 연세대학교 공학석사  
1999년 : University of Texas at Austin 공학박사  
1990년~1995년 : 국방과학연구소 연구원  
1999년~2000년 : 삼성종합기술원 전문연구원  
2000년~현재 : 광운대학교 전자융합공학과 교수  
<관심분야> 무선 LAN/PAN, 무선/모바일 보안, IoT, 인공지능/강화학습