https://doi.org/10.7840/kics.2023.48.2.258

Asynchronous Advantage Actor Critic 기반 그룹 임의접속 제어기술

김 수*, 장 한 승°

Group Random Access Control Scheme Based on Asynchronous Advantage Actor Critic

Su Kim*, Han Seung Jang°

요 약

본 논문은 대규모 그룹 IoT 기기들이 임의접속을 동시에 시도했을 때 발생하는 접속 과부하 문제를 해결하기 위해 Asynchronous Advantage Actor Critic (A3C) 기반의 임의접속 제어 기법을 제안한다. 기존 연구에서는 강화학습 기법인 DQN을 이용하여 접속 제어기술을 구현하던 것과는 달리, 본 연구에서는 강화학습 기법 중 A3C를 사용하는 접속제어 기술을 제안한다. 제안하는 A3C 기반의 그룹 임의접속 제어기술은 그룹의 모든 단말이 임의접속에 성공하는데 걸리는 시간 측면에서 최적 학습을 통해 일반적인 프리앰블 충돌 검출 방식과 빠른 프리앰블 충돌 검출 방식에 따른 이론적인 제어 성능에 도달함을 보여준다.

키워드 : 사물인터넷, 그룹 임의접속, 제어 정보, 강화 학습, 빠른 프리앰블 충돌 검출 방식

Key Words : Internet of Things, Group Random Access, Access Class Barring, Reinforcement Learning, Early Preamble Collision Detection

ABSTRACT

In the cellular Internet of Things (IoT) networks, the networks experience overload problems when a massive group of IoT devices attempts random access (RA) at the same time. In literature, existing schemes have designed RA control schemes based on Deep QLearning (DQN) with the restrictive choice of access class barring factor. Thus, in this paper, we propose the asynchronous advantage actor critic (A3C) based access control scheme for an exact prediction of access class barring factors between 0 and 1. The simulation results show that our proposed scheme outperforms the conventional access control schemes and it also reaches the theoretical performance in terms of total service time with the conventional and early preamble collision detection methods.

Ⅰ. 서 론

스마트폰, 노트북, 센서 등 수십억대 이상의 사물인터넷(Internet of Things, IoT) 기기가 이동통신 네트워크에 연결될 것으로 예상되는 가운데[1] 이와 같은 대규모의 단말들이 이동통신 기지국과의 초기 접속 및 동

※ 이 논문은 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (NRF-2021R1F1A1058795).
◆ First Author : Chonnam National University, School of Electrical, Electronic Communication, and Computer Engineering, ok96741016@gmail.com, 학생회원
° Corresponding Author : Chonnam National University, School of Electrical, Electronic Communication, and Computer Engineering, hsjang@jnu.ac.kr, 종신회원
논문번호 : 202209-224-A-RE, Received September 27, 2022; Revised November 30, 2022; Accepted December 5, 2022

기화를 위해서는 임의접속(Random Access, RA) 절차가 필수적이다. 또한, 대규모의 단말들을 그룹으로 관리하는 그룹 기반의 이동통신 기술이 IoT 환경에서 요구된다. 그룹 기반의 기술 중 하나인 그룹 임의접속의 절차는 총 5가지 단계로 이루어져 있다[2]. 대규모의 단말들이 속해있는 그룹이 동시에 임의접속을 시도할 경우 임의접속의 0단계가 매우 중요한데, 여기서 0단계는 t번째 시간 슬롯의 PRACH (Physical Random Access Channel)에서 기지국이 방송하는 제어 정보 $p^t \in [0,1]$에 따라 임의접속 1단계에 진입하는 단말의 수가 달라진다. 1단계에 진입하는 단말의 수가 많아질수록 임의접속을 시도하는 단말의 프리앰블 및 자원 충돌 경험 확률이 증가한다. 결과적으로 충돌로 인해 재접속의 횟수가 증가함에 따라 그룹의 모든 단말이 임의접속에 성공하는데 걸리는 시간이 증가하는 문제가 발생한다.

기존 연구에서는 위의 문제를 해결하기 위해 제어 정보 값을 최적화하는 연구들이 진행되었다. 먼저, 다양한 강화학습 기반의 임의접속 제어기술이 제안되었다[3,4]. 기존 연구에서는 Deep Q-Network (DQN) 학습 기법을 사용하였지만 제어 정보를 제한적인 선택지 내에서만 학습함에 따라 효과적인 접속제어에는 한계가 있었다. 두 번째로, 임의접속 과정 중 프리앰블 충돌 검출 방식에 따라 제어 정보를 최적화하였다. 여기서, 그룹 임의접속에 성공하는데 걸리는 총 서비스 시간의 관점에서 보면 일반적인 프리앰블 충돌 검출 방식의 총 서비스 시간보다 빠른 프리앰블 충돌 검출 방식의 총 서비스 시간이 더 짧다는 것이 이론적으로 밝혀졌다[5]. 본 논문에서는 강화학습 기법 중 Asynchronous Advantage Actor Critic(A3C) 모델을 활용하여 일반적인 프리앰블 충돌 검출 방식과 빠른 프리앰블 충돌 검출 방식 각각에 대해 최적 그룹 임의접속 제어 값을 도출하는 학습 모델을 설계한다. 마지막으로 최적화된 강화학습 모델을 통해 총 서비스 시간 측면에서 이론적인 성능에 도달함을 보인다.

Ⅱ. 본 론

본 논문에서는 총 서비스 시간의 최소화를 목표로 A3C 모델을 설계하여 그룹 임의접속을 최적 제어한다. 본래 임의접속은 각각의 단말에게 개별적으로 ID를 부여하여 기지국에 접속하게 하는데 극 다수의 단말이 기지국과 접속한다면 네트워크가 매우 혼잡해지고 기지국 입장에서 모든 단말을 일일이 관리하기에는 효율성이 떨어진다. 따라서 극 다수의 단말이 접속에 시도하는 경우 동일한 목적을 가진 단말들을 그룹으로 묶어 모두 같은 ID를 제공하고, 같은 ID를 가진 단말끼리 동시에 임의접속을 시도하는 그룹 임의접속 절차를 사용하여 효율을 높인다.

표 1. 시스템 변수 표
Table 1. Notation table for system model

Variable	Definition
N_{node}	The number of entire nodes in a group
T_{interval}	Time interval of random access procedure
N_{preamble}	The number of available preambles
N_{PUSCH}	The number of allocable PUSCH resources
q	Random value [0, 1] generated by a node
p^t	Access class barring factor at time slot t
N_{d}^t	The number of detected preambles at time slot t
N_{c}^t	The number of collided preambles at time slot t
N_{cf}^t	The number of collision-free preambles at time slot t
$N_{\mathrm{allocated}}^t$	The number of allocated preambles at time slot t
N_{s}^t	The number of successful nodes at time slot t
N_{b}^t	The number of backlogged nodes at time slot t
p_{conv}^t	Access class barring factor without consideration of PUSCH resources at time slot t
$\bar{p}_{\mathrm{JANG}}^t$	Access class barring factor with conventional preamble collision detection at time slot t
$\tilde{p}_{\mathrm{JANG}}^t$	Access class barring factor with early preamble collision detection at time slot t

본 연구에서는 $N_{\mathrm{node}} = 10{,}000$개의 임의접속을 요구하는 그룹 내 총 단말 수를 고려하고, 임의접속 절차의 한 주기는 $T_{\mathrm{interval}} = 50[\mathrm{ms}]$ 라고 가정한다. 이 주기 동안에는 기지국이 한 번의 임의접속 절차에서 성공과 실패한 단말 수를 정확히 파악할 수 있다. 마지막으로 그룹 내 모든 단말이 임의접속에 성공할 때까지 걸리는 시간을 총 서비스 시간으로 측정한다. 본 논문의 시스템 변수는 표 1에서 확인할 수 있다.

2.1 임의접속 과정 및 일반적인 프리앰블 검출 방식

임의접속 0단계에서는 t번째 시간 슬롯 제어 정보

가 p^t 일 때, 각 단말은 임의의 수 $q \in [0,1]$ 를 생성하고 $q \leq p^t$일 때 임의접속 1단계로 진입하여서 N_{preamble}개의 프리앰블 중 하나의 프리앰블을 임의로 선택하여 접속을 시도한다. 임의접속 2단계에서는 프리앰블 탐지와 자원할당이 진행된다. 한 대 이상의 단말이 선택한 프리앰블을 '탐지된 프리앰블'이라고 정의하고 이때 두 대 이상의 단말들이 동일한 프리앰블을 선택하는 상황을 '프리앰블 충돌(collision)'이라고 정의한다. 이와 반대로 하나의 단말이 독점적으로 하나의 프리앰블을 선택하는 상황을 '프리앰블 비충돌(collision-free)'이라고 정의한다. 따라서 '프리앰블 충돌'로 분류된 프리앰블 수를 N_{c}^t라고 하고, '프리앰블 비충돌'로 분류된 프리앰블 수를 N_{cf}^t 라고 할 때 탐지된 프리앰블 수 N_{d}^t는 다음과 같은 관계를 갖는다.

$$N_{\text{d}}^t = N_{\text{c}}^t + N_{\text{cf}}^t. \tag{1}$$

데이터 전송에 이용 가능한 자원 수를 N_{PUSCH}라고 할 때, 자원이 할당된 프리앰블의 수를 다음과 같이 나타낸다.

$$N^{t_{\text{allocated}}} = \min\left(N_{\text{d}}^t, N_{\text{PUSCH}}\right). \tag{2}$$

기지국 입장에서는 임의접속 2단계에서 탐지된 프리앰블 내에서 단말 간의 충돌 유무를 알 수 없어 자원 할당 시 모든 탐지된 프리앰블 중에서 한정된 자원 수만큼 선택하여 할당한다. 자원이 할당된 프리앰블을 선택한 단말에게는 탐지된 프리앰블 정보와 자원 할당 정보를 포함한 RAR (Random Access Response) 메시지를 송신하며, 임의접속 3단계에서는 할당받은 자원을 이용해 단말이 데이터를 전송한다. 마지막으로 임의접속 4단계에서는 기지국이 데이터를 해독하고 성공적으로 데이터를 해독하면 단말에게 ACK 메시지를 보낸다. 반대로 ACK 메시지를 받지 못한 단말들은 자신이 전송한 프리앰블이 '충돌 프리앰블'로 분류되었다는 사실을 알 수 있다. 이처럼 '일반적인 프리앰블 검출 방식(Conventional Preamble Collision Detection, C-PCD)'을 사용하면 단말은 임의접속 4단계에서 임의접속 성공 여부를 알 수 있어 시간 측면에서 낭비가 되고, '프리앰블 충돌'로 분류된 프리앰블에게도 자원이 할당되기 때문에 비효율적이다. 이 문제를 해결하기 위해 빠른 프리앰블 충돌 검출 방식을 사용한다.

2.2 빠른 프리앰블 충돌 검출 방식

단말이 프리앰블 충돌 여부를 알게 되는 시점이 임의접속 두 번째 단계일 경우 우리는 이를 '빠른 프리앰블 충돌 검출 방식(Early Preamble Collision Detection, E-PCD)'으로 부른다. 빠른 프리앰블 충돌 검출 방식은 임의접속 두 번째 단계에서 탐지된 프리앰블을 '프리앰블 비충돌'과 '프리앰블 충돌'로 빠르게 분류할 수 있다. '프리앰블 충돌'로 분류된 프리앰블에게 자원을 할당하여도 임의접속 세 번째 단계를 성공하지 못하므로 '프리앰블 비충돌'로 분류된 프리앰블에게만 자원을 할당하며 식 (2)는 다음과 같이 바뀌게 된다.

$$N_{\text{allocated}}^t = \min\left(N_{\text{cf}}^t, N_{\text{PUSCH}}\right). \tag{3}$$

그리고 자원을 할당받은 프리앰블을 선택한 단말에게만 RAR 메시지를 보냄으로써 RAR 메시지를 받지 않은 단말들은 곧바로 프리앰블 충돌 여부가 파악 가능하다. 이로써 단말의 접속 시간을 단축할 수 있으며, '프리앰블 충돌'로 분류된 프리앰블에게는 자원을 할당하지 않아 자원 낭비를 획기적으로 감소시킨다.

2.3 임의접속 제어 정보

총 서비스 시간을 최소화하기 위해서는 t번째 시간 슬롯에서 임의접속 네 번째 단계에서 ACK 메시지를 받은 임의접속 성공 단말의 수 N_{s}^t를 최대화하여야 한다. t번째 시간 슬롯에서 최대한 많은 단말이 임의접속을 성공하기 위해서는 임의접속 첫 단계에서 '프리앰블 비충돌'로 분류될 확률을 높여야 한다. 일반적인 프리앰블 검출 방식에서는 임의접속 두 번째 단계에서 탐지된 프리앰블에게 한정된 자원을 할당하게 된다. 식 (2)의 $N_{\text{allocated}}^t$ 중에 '프리앰블 비충돌'로 분류된 프리앰블 수의 비중을 높여야 한다. 이를 위해서는 프리앰블 충돌 검출 과정을 진행하기 전에 적절한 수의 단말이 임의접속을 시도하도록 단말의 수를 조정하여야 한다. 단말의 수를 조정하기 위해 임의접속 과정 0단계에서 임의접속 제어 정보(Access Control Barring factor)를 사용한다. 기존 기술에서는 현재 남아있는 단말 N_{b}^t 중에서 이용 가능한 프리앰블 수 N_{preamble}만큼의 단말이 임의접속할 수 있도록 제어 정보를 다음과 같은 식으로 설정한다.

$$p_{\text{conv}}^t = N_{\text{preamble}} / N_{\text{b}}^t. \tag{4}$$

그러나 p_{conv}^{t}는 이용 가능한 자원 수 N_{PUSCH}가 부족해지면 임의접속의 성공 확률이 떨어진다는 단점이 있음이 밝혀졌다[5]. 따라서 t번째 시간 슬롯에서의 제어 정보는 이용 가능한 프리앰블 수 N_{preamble} 뿐만 아니라 이용 가능한 자원 수 N_{PUSCH}를 함께 고려하여 비충돌 프리앰블 수 N_{cf}^{t}를 높여야 한다.

일반적인 프리앰블 충돌 검출 방식을 사용하였을 때 비충돌 프리앰블 수 N_{cf}^{t}는 탐지된 프리앰블 수 N_{d}^{t}와 이용 가능한 자원 수 N_{PUSCH}에 따라 결정된다. 이용 가능한 자원 수 N_{PUSCH} 보다 탐지된 프리앰블 수 N_{d}^{t}가 적거나 같을 때($N_{\mathrm{d}}^{t} \le N_{\mathrm{PUSCH}}$)는 탐지된 프리앰블에 자원을 전부 할당할 수 있으므로 프리앰블 충돌을 경험하지 않은 모든 단말들이 임의접속에 성공하게 된다. 반면에 이용 가능한 자원 수보다 탐지된 프리앰블 수가 더 많을 때($N_{\mathrm{d}}^{t} > N_{\mathrm{PUSCH}}$)는 모든 탐지된 프리앰블에게 자원을 할당할 수 없어 N_{PUSCH}개만이 자원 할당을 받게 된다. 이에 따라 임의접속 성공 단말 수 N_{s}^{t}를 최대화하기 위한 제어 정보 $\bar{p}_{\mathrm{JANG}}^{t}$를 다음과 같이 정리한다.

$$\bar{p}_{\mathrm{JANG}}^{t} = \begin{cases} \min\left\{1, -1/N_{\mathrm{b}}^{t}\ln(\gamma)\right\} & , \mathrm{if}\ D \le N_{\mathrm{PUSCH}} \\ \min\left\{1, \ln(\gamma)/N_{\mathrm{b}}^{t}\ln(\gamma)\right\} & , \mathrm{if}\ D > N_{\mathrm{PUSCH}}. \end{cases} \quad (5)$$

위 식에서 $\gamma = \left(1 - 1/N_{\mathrm{preamble}}\right)$이다. 그리고 평균적으로 탐지된 프리앰블 수 D는 $D = N_{\mathrm{preamble}}\left(1-(\gamma)^{-1/\ln(\gamma)}\right)$이다.

빠른 프리앰블 충돌 검출 방식을 사용하였을 때 N_{s}^{t}는 비충돌 프리앰블 수 N_{cf}^{t}와 할당 가능한 자원 수 N_{PUSCH}에 따라 결정된다. 빠른 프리앰블 충돌 검출 방식은 임의접속 두 번째 단계에서 충돌과 비충돌 프리앰블을 따로 분류할 수 있다. 따라서 비충돌 프리앰블 수 N_{cf}^{t}가 이용 가능한 자원 수 N_{PUSCH}보다 작거나 같을 때는($N_{\mathrm{cf}}^{t} \le N_{\mathrm{PUSCH}}$) 비충돌 프리앰블에게 자원을 전부 할당하고, 비충돌 프리앰블 수 N_{cf}^{t}가 이용 가능한 자원 수 N_{PUSCH}보다 더 많을 때($N_{\mathrm{cf}}^{t} > N_{\mathrm{PUSCH}}$)는 비충돌 프리앰블 중에서 N_{PUSCH}개만이 자원 할당을 받게 된다. 이에 따라 임의접속 성공 단말 수N_{s}^{t}를 최대화하기 위한 제어 정보 $\tilde{p}_{\mathrm{JANG}}^{t}$를 다음과 같이 정리할 수 있다.

$$\tilde{p}_{\mathrm{JANG}}^{t} = \begin{cases} \min\left\{1, -1/N_{\mathrm{b}}^{t}\ln(\gamma)\right\} & , \mathrm{if}\ S \le N_{\mathrm{PUSCH}} \\ \min\left\{1, \boldsymbol{W}(N_{\mathrm{PUSCH}} \cdot \gamma \cdot \ln(\gamma))/N_{\mathrm{b}}^{t}\ln(\gamma)\right\} & , \mathrm{if}\ S > N_{\mathrm{PUSCH}}, \end{cases} \quad (6)$$

여기서, $\boldsymbol{W}$는 람베르트 $\boldsymbol{W}$ 함수이고, 평균 비충돌 프리앰블 수 S는 $S = N_{\mathrm{preamble}}\left(-\frac{1}{\ln(\gamma)}\right)\left(\frac{1}{N_{\mathrm{preamble}}}\right)\left(1-\frac{1}{N_{\mathrm{preamble}}}\right)^{-1/\ln(\gamma)-1}$이다.

Ⅲ. Asynchronous Advantage Actor Critic 모델

Actor Critic은 정책 신경망(V)과 가치 신경망(C)으로 이루어져 있고 각각의 신경망에 상태를 입력으로 넣었을 때, 이산적인 값으로 정책과 가치를 예측하는 강화학습 방법이다. 행동으로 제어 정보를 설정했을 때, 한정적인 선택지 내에서 행동을 선택하는 것은 임의접속 성공 단말 수 N_{s}^{t}를 최대화하는 것에 제한적이다. 따라서 이산적으로 제어 정보 p^{t}를 선택하는 것이 아닌 연속적인 p^{t}값으로 제어 정보를 설정해야 한다.

연속적인 값으로 행동을 예측하는 Actor Critic 방법을 Advantage Actor Critic(A2C)이라고 한다. A2C 방법으로 하나의 에이전트가 연속적인 행동을 예측하려면 많은 상태와 그에 따른 행동을 탐험해야 하므로 오랜 시간이 필요하며, 이전 상태에 따른 행동 예측값에 영향을 많이 받는다. 이를 방지하기 위해 A2C 방법에서 멀티스레딩 기법을 적용하여 에이전트를 다수 생성 후 에이전트마다 동시에 각각의 상태에 따른 행동을 예측하여 현재 행동을 예측할 때, 이전 상태에 따른 행동 예측값의 영향을 덜 받게 한다. 이를 Asynchronous Advantage Actor Critic(A3C) 방법이라고 한다.

A3C의 환경 E는 프리앰블 충돌 검출 방식에 따라 일반적인 프리앰블 충돌 검출 방식 환경 $\bar{E}$와 빠른 프리앰블 충돌 검출 방식 환경 $\tilde{E}$를 구성한다. A3C 모델을 설계할 때 현재 상태(State), 행동(Action), 보상(Reward), 다음 상태(Next State)를 고려하며 다음과 같이 모델을 나타낼 수 있다[6].

$$\left\{S^{t}, A^{t}, R^{t}, S^{t+1}\right\}. \quad (7)$$

그리고 설계된 A3C 모델 알고리즘은 그림 1에서 확인할 수 있고, 의사 코드는 Algorithm 1에서 확인할 수 있다.

A3C 기법을 활용한 학습을 진행할 때 현재 남아있

는 단말 수 N_{b}^{t}를 현재 상태에 고려하는 것은 제어 정보 최적화에 많은 영향을 끼친다. 학습을 원활히 진행하기 위해서는 현재 상태에 남아있는 단말 수의 정보를 제공해야 하나, 본래 기지국 입장에서는 현재 상태에 남아있는 단말 수를 정확히 파악하기란 어려운 일이다. 따라서 학습 시 현재 상태에 남아있는 단말 수의 유무에 따른 A3C 모델을 구분해야한다. 따라서 남아있는 단말 수 적용 여부와 앞서 말한 두 가지의 환경에 따라 네 가지의 A3C 모델을 구성한다. 일반적인 프리앰블 충돌 검출 환경이면서 남아있는 단말 수를 아는 모델을 '모델 1,' 일반적인 프리앰블 충돌 검출 환경이면서 남아있는 단말 수를 모르는 모델을 '모델 2', 빠른 프리앰블 충돌 검출 환경이면서 남아있는 단말 수를 아는 모델을 '모델 3', 그리고 빠른 프리앰블 충돌 검출 환경이면서 남아있는 단말 수를 모르는 모델을 '모델 4'로 지칭한다.

3.1 상태(State)

현재 상태 S^{t}는 학습을 원활히 진행할 수 있도록 특정 변수를 정규화하고, 정규화 식은 다음과 같이 나타내며,

$$f(x)=\frac{x-x_{\min}}{x_{\max}-x_{\min}}, \tag{8}$$

현재 남아있는 단말 수 N_{b}^{t}가 적용되었을 경우와 적용되지 않았을 경우의 현재 상태 S^{t}는 아래의 식에서 확인할 수 있다.

$$S^{t}=\begin{cases}[f(N_{\mathrm{b}}^{t}),A^{t-1},f(\widehat{N}_{\mathrm{s}}^{t-1}),f(\widehat{N}_{\mathrm{d}}^{t-1}),R^{t-1}], & \text{if } N_{\mathrm{b}}^{t}\in S^{t}\\ [A^{t-1},f(\widehat{N}_{\mathrm{s}}^{t-1}),f(\widehat{N}_{\mathrm{d}}^{t-1}),R^{t-1}], & \text{else.}\end{cases} \tag{9}$$

현재 남아있는 단말 수가 적용되었을 경우 현재 남아있는 단말 수의 정규화 값 $f(N_{\mathrm{b}}^{t})$, 이전 행동 A^{t-1}, 이전 평균 단말 임의접속 성공 수의 정규화 값 $f(\widehat{N}_{\mathrm{s}}^{t-1})$, 이전 평균 프리앰블 탐지 수의 정규화 값 $f(\widehat{N}_{\mathrm{d}}^{t-1})$, 이전 보상 값 R^{t-1}으로 설정한다. 현재 남아있는 단말 수가 적용되지 않았을 경우 현재 상태 S^{t}는 이전 행동 A^{t-1}, 이전 평균 임의접속 성공 수의 정규화 값 $f(\widehat{N}_{\mathrm{s}}^{t-1})$, 이전 평균 프리앰블 탐지 수의 정규화 값 $f(\widehat{N}_{\mathrm{d}}^{t-1})$, 이전 보상 값 R^{t-1}으로 설정한다. 현재 남아있는 단말 수의 최댓값은 N_{node}이고, 이전 평균 임의접속 성공 수의 최댓값은 N_{PUSCH}이며, 이전 평균 프리앰블 탐지 수의 최댓값은 N_{Preamble}이고, 각각의 최솟값은 0이다. 이 최댓값과 최솟값을 식 (8)에 적용하여 각각의 변수들을 정규화할 수 있다.

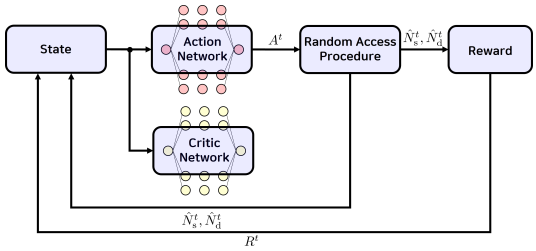

그림 1. Asyncronous Advantage Actor Critic(A3C) 알고리즘
Fig 1. Asyncronous Advantage Actor Critic(A3C) Algorithm

알고리즘 1. 제안한 모델의 의사 코드
Algorithm 1. Pseudocode for proposed model

Algorithm 1. Pseudocode for proposed model

Initialize E, S # Initialize Environment and State.
for episode = 1 to N_{Episode} do
  $t=0$ #Initialize the time slot.
  $N_{\mathrm{b}}^{t}=N_{\mathrm{node}}$ #Initialize the number of backlogged nodes.
  while $N_{\mathrm{b}}^{t}>0$ do
    $A^{t}=V(S^{t})$ #Predict the Action at time slot t.
    $v^{t}=C(S^{t})$ #Predict the Value at time slot t.
    $\widehat{N}_{\mathrm{s}}^{t},\widehat{N}_{\mathrm{d}}^{t}=\frac{1}{n}\sum_{i=1}^{n}E_{n}(A^{t})$ #Simulation n times in the Environment E with A^{t} and average the N_{s}^{t} and N_{d}^{t} separately.
    $\mathrm{Reward}=R(\widehat{N}_{\mathrm{s}}^{t},\widehat{N}_{\mathrm{d}}^{t})$
    #Set the Next State depending on the model.
    if Model = Model 1 or Model = Model 3 then
      $S^{t+1}=\left(f(N_{\mathrm{b}}^{t}),A^{t},f(\widehat{N}_{\mathrm{s}}^{t}),f(\widehat{N}_{\mathrm{d}}^{t}),R(\widehat{N}_{\mathrm{s}}^{t},\widehat{N}_{\mathrm{d}}^{t})\right)$
    elseif Model = Model 2 or Model = Model 4 then
      $S^{t+1}=\left(A^{t},f(\widehat{N}_{\mathrm{s}}^{t}),f(\widehat{N}_{\mathrm{d}}^{t}),R(\widehat{N}_{\mathrm{s}}^{t},\widehat{N}_{\mathrm{d}}^{t})\right)$
    end
    Update V weights.
    Update C weights.
    $N_{\mathrm{b}}^{t+1}=N_{\mathrm{b}}^{t}-\widehat{N}_{\mathrm{s}}^{t}$ #Update the number of backlogged nodes.
    $t=t+1$ # Update the time slot t.
  end
end

3.2 행동(Action)

본래 정책 신경망에서는 다수의 행동이 있을 때, 그 행동들이 나올 확률값을 예측하고, 반환된 확률로 행동을 선택하는 모델이었다. 그러나 본 연구의 모델에서

행동은 0과 1 사이의 연속적인 제어 정보 p^t이기 때문에 정책 신경망에서 예측한 0과 1사이의 확률값이 그대로 제어 정보 p^t가 된다. 예측된 제어 정보는 구성된 학습 모델에 따라 p^t_{Model1}, p^t_{Model2}, p^t_{Model3}, 그리고 p^t_{Model4}로 나타낸다.

3.3 임의접속 시뮬레이션

본래 A3C 기법에서는 t번째 슬롯에서 예측된 Actor 값 A^t로 한 번의 환경을 실험하고 나온 결과인 보상을 가중치 업데이트에 이용한다. 하지만 본 논문의 환경에서는 확률적으로 변화하는 임의접속 결과에 따라 불안정하게 학습이 진행될 수 있는 문제점을 가지고 있다. 따라서 t번째 슬롯에서 A^t을 사용하여 임의접속 과정을 n번 반복 시뮬레이션 함으로써 평균 임의접속 성공 수 $\widehat{N}_{\text{s}}^t$과 평균 프리앰블 탐지 수 $\widehat{N}_{\text{d}}^t$를 얻는다. 이를 통해 N_{b}^{t+1}는 $N_{\text{b}}^{t+1} = N_{\text{b}}^t - \widehat{N}_{\text{s}}^t$로 갱신된다.

3.4 보상(Reward)

보상으로 현재 예측된 행동 값을 평가할 수 있다. 보상이 높을수록 현재 예측된 제어 정보가 최적화되었다는 의미이다. 보상은 평균 그룹 임의접속 성공 수 $\widehat{N}_{\text{s}}^t$의 유무에 따라 다음과 같이 두 가지로 분류하였다. 첫 번째로, 성공한 단말이 한 개 이상일 때 현재 그룹 임의접속을 시도한 단말의 수가 적절함을 의미하며, $\widehat{N}_{\text{s}}^t$을 최대화하는 것이 목표이므로 $\widehat{N}_{\text{s}}^t$ 값에 비례하여 보상 값을 설정했다. 두 번째로, 그룹 임의접속에 성공한 단말이 없을 경우는 단말이 들어온 수에 따라 제어 정보를 잘못 예측한 경우이다. 따라서 평균 탐지된 프리앰블 수 $\widehat{N}_{\text{d}}^t$에 따라 음수 값으로 비례하도록 설정하였다. 여기서 α와 β는 하이퍼 파라미터로 최적화를 위해 설정된 상수다.

$$R^t = \begin{cases} \log\left(\widehat{N}_{\text{s}}^t / \widehat{N}_{\text{d}}^t \times \alpha\right), & \text{if } \widehat{N}_{\text{s}}^t > 0 \\ -\widehat{N}_{\text{d}}^t \times \beta, & \text{else.} \end{cases} \tag{9}$$

Ⅳ. 실 험

본 논문에서 사용한 실험 변수 및 모델 구성 방식을 표 1에 정리하였다. 그룹 내의 총 단말 수는 N_{node} = 10,000대로 가정하였다. 이용 가능한 자원 수 N_{PUSCH}를 10, 20, 30, 40, 50을 가정하여 학습을 진행하였고, 제어 정보 p^t를 0과 1 사이의 확률값으로 예측하기 위해 행동 신경망 모델 마지막 층의 활성 함수를 시그모이드(Sigmoid) 함수로 지정하였다. 기지국이 성공한 단말 수를 아는 모델 1과 3의 은닉층 수가 성공한 단말 수를 모르는 모델 2와 4의 은닉층 수보다 더 적은 이유는 현재 남아있는 단말 수 N_{b}^t를 입력으로 넣었을 때 가중치 업데이트에 중요한 역할을 하여 깊게 층을 쌓지 않아도 그림 2와 같이 최적화된 결과에 빠르게 수렴하기 때문이다. 현재 남아있는 단말 수 N_{b}^t를 상태에 포함하지 않은 모델 2와 4의 학습을 원활히 진행하기 위해서는 은닉층을 깊게 쌓아야 한다. 하지만 은닉층 수가 깊으면 깊을수록 기울기 소실 문제가 발생한다. 이는 비선형 활성 함수를 사용함으로써 해결할 수 있다. 따라서 모델 2와 4에서는 Rectified Linear Unit

표 2. 실험 변수
Table 2. Parameter

Parameter	Value
The number of all nodes(N_{node})	10,000
The number of available preambles(N_{preamble})	64
The number of allocable PUSCH resources(N_{PUSCH})	10, 20, 30, 40, 50
The number of episodes(N_{episode})	25
Model 1 Hyperparameter(α, β)	1.2, $0.1/N_{\text{PUSCH}}$
Model 2 Hyperparameter(α, β)	1.3, 0.15625
Model 3 Hyperparameter(α, β)	1.2, $0.1/N_{\text{PUSCH}}$
Model 4 Hyperparameter(α, β)	1.3, 0.078125
Actor hidden layer for Model 1 & 3 (The number of units)	3 (128, 128, 128)
Critic hidden layer for Model 1 & 3 (The number of units)	3 (128, 128, 128)
Actor hidden layer for Model 2 & 4 (The number of units)	5 (128, 128, 128, 128, 128)
Critic hidden layer for Model 2 & 4 (The number of units)	5 (128, 128, 128, 128, 128)
Actor network activation for Model 1 & 3	(ReLU, ReLU, ReLU, Sigmoid)
Critic network activation for Model 1 & 3	(ELU, ELU, ELU, ELU)
Actor network activation for Model 2 & 4	(ReLU, ReLU, ReLU, ReLU, ReLU, Sigmoid)
Critic network activation for Model 2 & 4	(ELU, ELU, ELU, ELU, ELU, ELU)

(ReLU)를 사용한다. Critic 은닉층에서도 기울기 소실 문제를 해결하면서 정확한 가치를 반환하기 위해 비선형 함수인 Exponential Linear Unit (ELU)을 사용한다.

그림 3에서 t번째 시간 슬롯에서 각각의 이용 가능한 자원 수에 따라 식 (4), (5), 그리고 (6)에서 계산된 이론적인 제어 정보 p_{conv}^{t}, $\bar{p}_{\mathrm{JANG}}^{t}$, 그리고 $\tilde{p}_{\mathrm{JANG}}^{t}$ 값에 따른 그룹 임의접속의 총 서비스 시간을 확인할 수 있다. 사용 가능한 자원이 적으면 적을수록 자원 양을 고려하지 않은 p_{conv}^{t}을 사용한 그룹 임의접속의 총 서비스 시간보다 자원 양을 고려한 $\bar{p}_{\mathrm{JANG}}^{t}$과 $\tilde{p}_{\mathrm{JANG}}^{t}$을 사용했을 때 그룹 임의접속의 총 서비스 시간이 현저히 줄어드는 것을 확인할 수 있다. 이는 프리앰블 수에 따른 제어 정보를 가지게 되었을 때 자원이 적어질수록 비충돌 프리앰블에게 자원을 할당하는 비율보다 충돌 프리앰블에게 자원을 할당하는 비율이 높아지기 때문에 자원 낭비뿐만 아니라 접속 시간이 지연됨을 알 수 있다. 이론적으로 $\bar{p}_{\mathrm{JANG}}^{t}$과 $\tilde{p}_{\mathrm{JANG}}^{t}$을 사용했을 때 이용 가능한 자원 수만큼 단말들을 접속하도록 제어했기 때문에 비충돌 프리앰블에게 자원을 할당하는 비율을 높일 수 있고, 이에 따라 자원 낭비를 줄일 수 있을 뿐만 아니라 접속 시간이 단축되는 것을 확인할 수 있다. 이를 제안하는 모델에서 예측된 제어 정보 p_{Model1}^{t}, p_{Model2}^{t}, p_{Model3}^{t}, 그리고 p_{Model4}^{t}에 따른 그룹 임의접속의 총 서비스 시간과 성능을 비교한다. 일반적인 프리앰블 충돌 검출 방식 환경일 때 p_{Model1}^{t}과 p_{Model2}^{t}의 최적값을 활용한 그룹 임의접속의 총 서비스 시간 성능은 이론값 $\bar{p}_{\mathrm{JANG}}^{t}$에 도달함을 알 수 있다. 빠른 프리앰블 충돌 검출 방식 환경일 때 p_{Model3}^{t}과 p_{Model4}^{t}의 최적값을 활용한 그룹 임의접속의 총 서비스 시간 성능은 이론값 $\tilde{p}_{\mathrm{JANG}}^{t}$에 도달함을 알 수 있다. 이는 임의접속에 대한 구체적인 이론과 수학적 지식이 없는 상황에서도 임의접속 과정에서 기지국이 얻을 수 있는 기본적인 상태 정보와 상태 정보를 입력값으로 하는 보상함수를 세울 수 있다면 다양한 자원 제약 환경에서 이론적인 최적 성능에 도달하는 그룹 제어 모델을 예측할 수 있음을 의미한다.

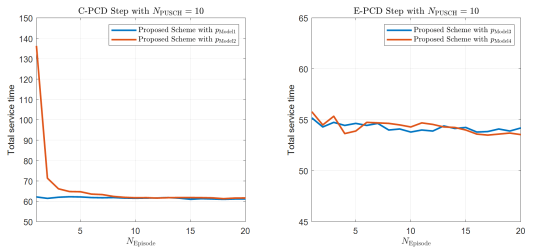

그림 2. 스텝에 따른 총 서비스 시간 수렴 속도 ($N_{\mathrm{PUSCH}}=10$)
Fig 2. The convergence speed of total service time for steps($N_{\mathrm{PUSCH}}=10$)

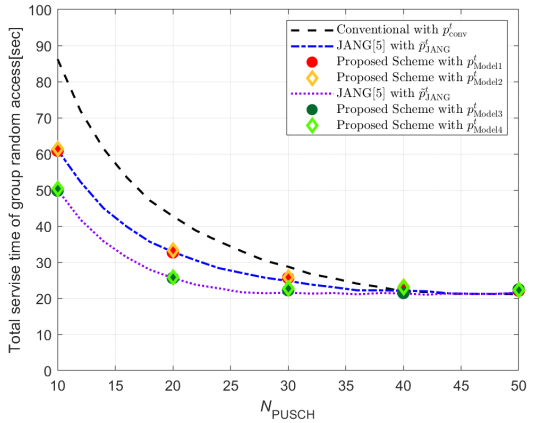

그림 3. 이용 가능한 자원 수에 따른 그룹 임의접속의 총 서비스 시간
Fig 3. The total service time of group random access for the number of available resources

V. 결 론

본 논문에서는 대규모 그룹 IoT 기기들이 임의접속을 동시에 시도했을 때 발생하는 접속 과부하 문제를 강화학습과 프리앰블 충돌 검출 방식 측면에서 해결하고자 했다. 강화학습 측면에서는 연속적인 제어 정보를 최적화할 수 있는 A3C 기법을 사용했다. 프리앰블 충돌 검출 방식 측면에서는 일반적인 프리앰블 충돌 검출 방식과 빠른 프리앰블 충돌 검출 방식을 각각 적용하였다. A3C 기법의 환경에 두 가지의 프리앰블 충돌 검출 방식을 사용하고, 상태에 남아있는 단말 수의 유무를 적용하여 제어 정보를 최적화하였다. 결과적으로 A3C 기반 그룹 임의접속의 총 서비스 시간이 일반적인 프리앰블 충돌 검출 방식과 빠른 프리앰블 충돌 검출 방식 측면에서 이론적인 총 서비스 시간에 도달하는 것을 확인했다. 이로써 그룹 임의접속 상황에서 기지국이 한정된 정보로 최적화된 제어 정보를 도출하여 단말의 그룹 임의접속 성공 수를 최대화하여 접속 과부하 문제를 해결할 수 있음을 확인하였다.

References

[1] I. Yaqoob, et al., “Internet of things architecture: Recent advances, taxonomy, requirements, and open challenges,” in *IEEE Wireless Commun.*, vol. 24, no. 3, pp. 10-16, Jun. 2017. (https://doi.org/10.1109/MWC.2017.1600421)

[2] H. S. Jang, H. Jin, B. C. Jung, and T. Q. S. Quek, “Versatile access control for massive iot: Throughput, latency, and energy efficiency,” in *IEEE Trans. Mob. Comput.*, vol. 19, no. 8, pp. 1984-1997, Aug. 2020. (https://doi.org/10.1109/TMC.2019.2914381)

[3] D. Pacheco-Paramo, et al., “Deep reinforcement learning mechanism for dynamic access control in wireless networks handling mMTC,” *Ad Hoc Netw.*, vol. 94, no. 101939, 2019. (https://doi.org/10.1016/j.adhoc.2019.101939)

[4] N. Jiang, Y. Deng, and A. Nallanathan, “Deep reinforcement learning for discrete and continuous massive access control optimization,” *2020 IEEE ICC*, pp. 1-7, 2020. (https://doi.org/10.1109/ICC40277.2020.9149055)

[5] H. S. Jang, B. C. Jung, and D. K. Sung, “Dynamic access control with resource limitation for group paging-based cellular IoT systems,” in *IEEE Internet of Things J.*, vol. 5, no. 6, pp. 5065-5075, Dec. 2018. (https://doi.org/10.1109/JIOT.2018.2873429)

[6] V. Mnih, et al., “Asynchronous methods for deep reinforcement learning,” *Int. Conf. Mach. Learn.* PMLR, pp. 1928-1937, 2016. (http://proceedings.mlr.press/v48/mniha16.html?ref=https://githubhelp.com)

김 수 (Su Kim)

2022년 2월 : 전남대학교 전기·전자통신·컴퓨터공학부 졸업

2022년 3월~현재 : 전남대학교 전자통신공학과 석사과정

<관심분야> 사물인터넷, 인공지능, 이동통신

장 한 승 (Han Seung Jang)

2012년 2월 : 전남대학교 전자공학 학사

2014년 2월 : 한국과학기술원 전기 및 전자공학 석사

2017년 8월 : 한국과학기술원 전기 및 전자공학 박사

2019년 3월~2022년 3월 : 전남대학교 전기·전자통신·컴퓨터공학부 조교수

2022년 4월~현재 : 전남대학교 전기·전자통신·컴퓨터공학부 부교수

<관심분야> 사물인터넷, 인공지능, 에너지 ICT

[ORCID: 0000-0002-9024-8952]