The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)
Vol. 23, No. 6, pp.61-67, Dec. 31, 2023. pISSN 2289-0238, eISSN 2289-0246

https://doi.org/10.7236/JIIBC.2023.23.6.61
JIIBC 2023-6-10

MEC를 활용한 커넥티드 홈의 DRL 기반 태스크 오프로딩 기법

Task offloading scheme based on the DRL of Connected Home using MEC

임덕선*, 손규식*

Ducsun Lim*, Kyu-Seek Sohn*

요 약 5G의 도래와 스마트 디바이스의 급격한 증가는 멀티 액세스 엣지 컴퓨팅(MEC)의 중요성을 부각시켰다. 이런 흐름 속에서, 특히 계산 집약적이고 지연시간에 민감한 애플리케이션의 효과적인 처리가 큰 관심을 받고 있다. 본 논문에서는 이러한 도전 과제를 해결하기 위해 확률적인 MEC 환경을 고려한 새로운 태스크 오프로딩 전략을 연구한다. 먼저 동적인 태스크 요청 빈도와 불안정한 무선 채널 상태를 감안하여 차량의 전력 소모와 지연시간을 최소화하는 방안을 제시한다. 그리고 심층 강화학습(DRL) 기반의 오프로딩 기법을 중심으로 연구를 진행하였고, 로컬 연산 및 오프로딩 전송 전력 사이의 최적의 균형을 찾기 위한 방법을 제안한다. Deep Deterministic Policy Gradient (DDPG)와 Deep Q-Network (DQN) 기법을 활용하여 차량의 전력 사용량과 큐잉 지연시간을 분석하였다. 이를 통해 차량 기반의 MEC 환경에서의 최적의 성능 향상 전략을 도출 및 검증하였다.

Abstract The rise of 5G and the proliferation of smart devices have underscored the significance of multi-access edge computing (MEC). Amidst this trend, interest in effectively processing computation-intensive and latency-sensitive applications has increased. This study investigated a novel task offloading strategy considering the probabilistic MEC environment to address these challenges. Initially, we considered the frequency of dynamic task requests and the unstable conditions of wireless channels to propose a method for minimizing vehicle power consumption and latency. Subsequently, our research delved into a deep reinforcement learning (DRL) based offloading technique, offering a way to achieve equilibrium between local computation and offloading transmission power. We analyzed the power consumption and queuing latency of vehicles using the deep deterministic policy gradient (DDPG) and deep Q-network (DQN) techniques. Finally, we derived and validated the optimal performance enhancement strategy in a vehicle based MEC environment.

Key Words : Deep Reinforcement Learning, Deep Q-Network, Deep Deterministic Policy Gradient, Markov Decision Process, Connected home

*정회원, 한양사이버대학교 해킹보안학과
접수일자 2023년 10월 18일, 수정완료 2023년 11월 18일
게재확정일자 2023년 12월 8일

Received: 18 October, 2023 / Revised: 18 November, 2023 / Accepted: 8 December, 2023
*Corresponding Author: kssohn@hycu.ac.kr
Dept. of Hacking and Security, Hanyang Cyber University, Korea

Ⅰ. 서 론

5G 시대가 도래하면서 스마트 디바이스의 수가 기하급수적으로 증가하였다. 이로 인해 보안 카메라, 스마트 가전 기기, 홈 자동화 시스템 같은 계산 집약적이고 지연시간에 민감한 애플리케이션들이 대두되었다[1]. 그러나 스마트 디바이스의 제한된 배터리와 컴퓨팅 자원으로 인해 이러한 애플리케이션들의 효율적인 실행은 어려웠다. 멀티 액세스 엣지 컴퓨팅(MEC)은 이 문제를 해결하는 유망한 기술로 주목받고 있다[2].

MEC는 사용자에 가까운 지점에서 클라우드와 유사한 컴퓨팅 기능을 제공하여 지연시간을 줄이고 처리 효율을 향상시킨다[3]. 스마트 디바이스는 MEC를 활용해 계산 부하가 큰 태스크를 근접한 MEC 서버로 오프로딩하며, 이로 인해 지연시간과 배터리 소모를 줄이게 된다.

특히 커넥티드 홈 같은 복잡한 환경에서는 MEC가 중요한 역할을 한다[4]. 실시간 응답이 필수적인 이 환경에서 MEC는 높은 네트워크 신뢰성과 낮은 처리 지연시간을 제공한다. 최근의 연구에서는 이러한 문제를 해결하기 위한 강화학습 기반의 태스크 오프로딩 방법에 주목하고 있다.

MEC는 큰 장점을 가지고 있지만, 아직도 해결해야 할 과제들이 있다. 실시간 애플리케이션들의 엄격한 지연시간 요구 사항과 다양한 요인들이 영향을 주기 때문에, 오프로딩 최적화는 복잡한 문제로 남아있다[5]. 특히 무선 채널의 변동성과 스마트 디바이스의 위치 변화를 고려하면, 오프로딩 결정은 더욱 어려워진다.

태스크 오프로딩 및 자원 할당 연구는 지연시간 최소화, 에너지 소비 감소, 또는 두 요소의 균형을 중점으로 한다[6-8]. 그러나 많은 연구들이 MEC의 동적인 특성을 충분히 고려하지 않는다. 이 문제를 해결하기 위해 강화학습 기반의 기법들이 제안되었다. Deep Q-network (DQN)와 Deep Deterministic Policy Gradient (DDPG) 기법은 이 중 주목받는 방법들이다[9-11].

본 논문에서는 커넥티드 홈 환경에서 DRL 기반의 오프로딩 기법을 제안하며, DDPG와 DQN 기반 기법의 성능을 지연시간 및 에너지 측면에서 비교 분석한다. 본 논문은 다음과 같이 구성되어 있다. Ⅱ장에서는 시스템 모델에 대해 설명하고, Ⅲ장에서는 제안하는 알고리즘에 관해 기술하였다. Ⅳ장에서는 실험 및 성능 비교 분석을 확인하고 Ⅴ장에서는 결론을 통해 마무리 짓는다.

Ⅱ. 시스템 모델

본 논문에서는 MEC 시스템을 모델을 설계하였다. 여기서 MEC는 M개로 구성되며, 스마트 디바이스의 수는 N개로 가정한다. 각 MEC는 BS(Base Station)에 배치되어 있으며, 스마트 디바이스는 가장 가까운 MEC나 신호 세기가 높은 MEC m을 오프로딩 대상으로 선택한다. 고성능의 MEC는 동시에 여러 태스크를 처리할 수 있으므로, 오프로드된 태스크들은 동시에 처리되기로 가정한다.

각 스마트 디바이스는 $t \in T$에 평균 태스크 도착률 η에 따라 $K_n(t)$만큼의 태스크를 생성한다. 이 태스크들은 분할 가능하므로, 필요에 따라 태스크의 일부만 MEC로 오프로딩 될 수 있다. $\delta_{n,m}(t)$는 스마트 디바이스 n과 MEC m 사이의 통신 효율은 채널 이득을 통해 측정된다. 여기서 ξ^2는 잡음 전력, B는 전송 대역폭을 의미한다. $\delta_{n,m}(t)$와 스마트 디바이스의 송신 전력 $p_n^{tran}(t)$에 따라 스마트 디바이스 n이 MEC m으로 태스크를 오프로딩하는 전송 속도는 다음과 같은 수식 (1)로 정의된다.

$$D_n^{tran}(t) = B\log_2\left(1 + \frac{p_n^{tran}(t)\delta_{n,m}(t)}{\xi^2}\right) \quad (1)$$

$f_n(t) \in [0, F_n]$는 스마트 디바이스 n의 로컬 처리를 위해 할당된 CPU 주파수를 나타내며, C_n은 해당 디바이스에서 처리되는 태스크 당 필요한 CPU 사이클 수이다. 스마트 디바이스에서 로컬로 처리될 수 있는 데이터의 양은 수식 (2)로 나타낸다.

$$D_n^{loc}(t) = \frac{f_n(t)}{C_n} \quad (2)$$

이 데이터를 로컬에서 처리할 때, 소비되는 에너지는 다음의 수식(3)으로 표현될 수 있다. 여기서 μ는 칩 아키텍처 특성에 따른 에너지 효율 계수를 나타낸다.

$$p_n^{loc}(t) = \mu(f_n(t))^3 \quad (3)$$

타임슬롯 t에 태스크가 $K_n(t)$만큼 발생한다. 이 때 스마트 디바이스에서 처리되는 태스크의 양을 수식 (1)과 (2)로 표현하면, 해당 디바이스에서 대기열의 길이는 수식 (4)로 정의될 수 있다.

$$\Phi_n(t+1) = \max[\Phi_n(t) - D_n^{loc}(t) + D_n^{tran}(t)]^+ + K_n(t) \quad (4)$$

본 논문의 주목적은 스마트 디바이스의 태스크 처리량을 극대화하고 에너지 소모를 최소화함으로써 큐잉 지연 시간을 최소화하는 것이 목표이다. 큐잉 이론에서는 여러 가지 법칙과 모델을 통해 시스템의 성능을 분석하고 최적화할 수 있다. M/M/1 큐 모델을 도입하여 서비스 속도, 도착률 및 시스템의 효율성을 평가할 수 있다[12]. M/M/1 모델은 단일 서비스 제공자와 무한 크기의 대기열을 가진 시스템을 나타낸다. 이 모델을 이용하면 다양한 시스템 지표를 도출하고 분석할 수 있다. 따라서 $\omega \in [0,1]$를 사용하여 가중치 합계로 정의하고, 제약 조건들을 고려하여 수식으로 표현하면 다음 (5)와 같이 정의된다.

$$\begin{aligned} \min\ & \omega(p_n^{loc}(t) + p_n^{tran}(t)) + (1-\omega)\Phi_n(t) \quad (5) \\ s.t\ & f_n(t) \in [0, F_n],\ \forall t \in T \\ & p_n^{tran}(t) \in [0, p_n^{tran}],\ \forall t \in T \end{aligned}$$

여기서 ω는 에너지 소모와 성능 지표 사이의 균형을 맞추기 위한 가중치로, 두 지표 간의 상대적 중요도를 결정한다. 수식은 지연시간과 에너지 소모량 간의 트레이드 오프를 최적화하는 방향으로 구성되어 있다. 따라서 ω의 조절을 통해 어느 한 측면을 더 중요하게 여길지 결정할 수 있다.

Ⅲ. 제안 방법

본 논문에서 MEC를 활용하여 최적의 태스크 오프로딩을 활용하는 알고리즘을 제안한다. 제안하는 기법은 스마트 디바이스에서 최적의 연산 자원을 할당하여 로컬 실행을 진행하고, 오프로딩을 통해 최적의 전송 전력을 배정함으로써 지연시간과 에너지 소모를 최소화하는 DRL 기반의 태스크 오프로딩 기법을 제안한다.

1. 심층 강화학습

DQN은 Q-Learning의 Q-table을 딥 러닝 기반의 신경망으로 치환해 학습하는 방법론이다. 이 신경망을 이용하여 Q함수를 근사하며, 이를 $Q^*(s,a|\theta)$로 표현하자면, θ는 메인 네트워크의 파라미터를 의미한다.

DQN[9]의 핵심은 타깃 네트워크 $Q(s',a'|\theta')$와 메인 네트워크 $Q(s,a|\theta)$ 사이의 평균 제곱 오차(mean square error, MSE)를 최소화하는 방향으로 학습을 진행하는 것이며, 이는 수식 (6)으로 표현된다.

$$L(\theta) = E\left[\left(r + \gamma \max_{a'} Q(s',a'|\theta') - Q(s,a|\theta)\right)^2\right] \quad (6)$$

또한, ε-그리디 전략을 통해 환경 탐색 시 무작위성을 주어 가끔식 보상을 기준으로 하지 않는 탐사를 진행한다.

DDPG는 DQN과 액터-크리틱(Actor-critic)을 융합한 알고리즘이다[11]. 이 방법은 액터 네트워크 $\varsigma(s|\theta^{\varsigma})$와 Q 함수를 근사하는 크리틱 네트워크 $Q(s,a|\theta^Q)$ 두 개의 네트워크를 동시에 학습한다. 액터 네트워크는 최적의 행동을 추출하는데 사용되며, 크리틱 네트워크는 해당 행동의 가치를 평가한다. DQN의 방식처럼 DDPG는 타깃 네트워크를 사용하여 학습의 안정성을 확보한다. 따라서 액터와 크리틱 각각에 $\varsigma'(s|\theta^{\varsigma'})$, $Q'(s,a|\theta^{Q})$를 생성하여 목표 값을 계산한다. 액터 네트워크는 정책 경사(policy gradient) 방법을 통해 학습되며, 이는 수식 (7)과 같이 표현된다.

$$\nabla_{\theta^{\varsigma}} \vartheta = E\left[\nabla_a Q(s,a|\theta^Q) \nabla_{\theta^{\varsigma}} \varsigma(s|\theta^{\varsigma})\right] \quad (7)$$

액터 네트워크의 학습은 액션 가치 함수의 경사와 정책의 경사를 곱한 형태로 진행된다. 이 과정을 통해, 정책의 매개변수가 최적화되어 기대되는 보상을 극대화하게 된다. 한편, 크리틱 네트워크는 손실 함수를 기반으로 학습하며, 이는 수식 (8)으로 표현되어, 이 함수값을 최소화하는 방향으로 매개변수가 업데이트된다.

$$L(\theta^Q) = E\left[\left(r + \gamma Q(s', \varsigma'(s|\theta^{\varsigma})|\theta^Q) - Q(s,a|\theta^Q)\right)^2\right] \quad (8)$$

여기서 $Q(s,a|\theta^Q)$는 현재 상태와 행동에 기반한 예상 보상을 나타내고, $r + \gamma Q(s', \varsigma'(s|\theta^{\varsigma})|\theta^Q$는 다음 상태에서의 할인된 예상 보상을 나타낸다. 이 두 값의 차이의 제곱을 손실로 사용하여 크리틱 네트워크를 학습시킨다

Off-policy 전략을 적용한 DDPG는 탐험과 학습 메커니즘을 분리하여 처리한다[13]. 따라서 탐험 정책 ς'는 액터 정책에 노이즈 $\triangle\varsigma$를 추가함으로써, 수식 (9)로 정의된다.

$$\varsigma'(s) = \varsigma(s|\theta^{\varsigma}) + \triangle\varsigma \quad (9)$$

주어진 상태 s에서 어떻게 정책을 업데이트할 것인지 결정하게 되며, $\triangle\varsigma$는 기존의 정책 $\varsigma(s|\theta^{\varsigma})$에 얼마의 변화를 부여할 것인지를 나타내는 파라미터이다.

데이터 간의 상관성을 최소화하여 심층 신경망의 학습을 용이하게 만들기 위해 리플레이 버퍼(replay buffer) 기법을 도입한다[14]. 타깃 네트워크의 업데이트는 수식 (10) 및 (11)을 따라 수행된다.

$$\theta^{Q} \leftarrow \tau\theta^{Q} + (1-\tau)\theta^{Q} \quad (10)$$

$$\theta^{\varsigma'} \leftarrow \tau\theta^{\varsigma} + (1-\tau)\theta^{\varsigma'} \quad (11)$$

수식을 활용하여 타깃 값의 빠른 변화를 완화시켜, 학습 과정의 안정성을 향상시킨다.

2. Markov Decision Process

본 논문에서는 각각의 스마트 디바이스가 에이전트로써 독립적으로 오프로딩 정책을 학습한다. 이를 통해 지연시간 및 에너지 소모를 최소화하는 최적의 정책을 찾아낸다. 이러한 과정을 MDP(markov decision process)로 표현하면 아래와 같이 정의할 수 있다.

가. 상태 공간(State space)

스마트 디바이스의 MEC 오프로딩 결정의 기반이 되는 상태는 디바이스의 배터리 잔량, 대기열에 있는 태스크의 개수, 채널의 상태 정보, 그리고 태스크의 연산량, 그리고 데이터 전송량을 포함한다. 이러한 상태 정보는 (12)와 같이 표현된다.

$$s_t = (p_t, q_t, c_t, w_t) \quad (12)$$

이 정보들은 오프로딩의 결정 및 방식에 중요한 영향을 미친다.

나. 행동 공간(Action space)

Action의 두 가지 요소로 구성되어 있다. 행동의 정의는 (13)과 같다.

$$a_t = \delta, P_{tran} \quad (13)$$

첫 번째 요소에서는 로컬에서의 태스크 실행 비율인 δ를 나타내며, 두 번째 요소는 전송 전력 수준인 P_{tran}이다. δ는 0에서 1 사이의 값을 가진다. 값이 1일 경우 태스크를 로컬에서만 실행하며, 0일 경우 태스크를 MEC 서버로 오프로딩한다. P_{tran}은 최소 및 최대 전력 사이의 값을 가진다.

다. 보상(Reward)

보상 함수는 MEC 오프로딩의 성능을 측정하는 기준으로 사용되며, 그 성능은 여러 가지 요인들을 통합적으로 반영한다.

$$r_t = \alpha(p_{t+1} - p_t) - \beta q_{t+1} + \varpi c_t + 1 \quad (14)$$

여기서 α, β, ϖ는 각 항목의 상대적 중요도를 나타내는 가중치를 의미한다. $p_{t+1} - p_t$는 전력 소모량의 변화를 나타내며, 얼마나 전력 효율성이 향상되었는지를 평가한다. q_{t+1}은 다음 시간 단계에서의 대기열 길이나 지연시간을 나타내며, 태스크의 처리 속도를 중점적으로 반영한다. 마지막으로 c_{t+1}은 다음 시간 단계에서의 통신 채널의 상태를 나타내는 보상으로, 오프로딩의 성공 확률과 통신의 효율성을 고려한다.

3. 제안하는 오프로딩 알고리즘

DQN과 DDPG는 심층 강화 학습 알고리즘에서 주요한 변형 알고리즘들로 꼽히며, 복잡한 문제 영역에서의 효율적인 학습을 지원한다[15-16]. MEC 오프로딩과 같은 상황에서는 상태, 행동, 그리고 보상의 정확한 설정을 통해 이러한 알고리즘들을 적절하게 적용하는 것이 중요하다. 여기서 제시하는 것은 DQN 및 DDPG 기반으로 한 태스크 오프로딩 알고리즘 의사 코드이다.

DQN 학습 과정은 주요한 단계들로 구성된다. 먼저, 학습을 위한 목표로 Q-network와 목표로 사용되는 target Q-network라는 두 개의 신경망을 초기화한다. Q-network는 학습 과정에서 지속적으로 업데이트되는 반면, target Q-network는 정해진 간격으로 Q-network의 가중치를 받아와서 갱신된다. 각 에피소드에서는 ϵ-greedy 정책을 통해 행동을 결정한다. 이 정책은 탐색과 활용의 균형을 보장한다. 특정 행동을 취한 후, 결과로 받은 보상과 다음 상태를 파악하여 experience replay memory에 저장한다. 그 후, replay memory에

서 무작위 샘플링을 통해 데이터의 minibatch를 구성하고, 이를 바탕으로 가져와 Q-value의 갱신 대상을 설정한다. Q-network는 이 정보를 바탕으로 손실 함수를 최소화하는 방향으로 갱신되고, 주기적으로 target Q-network도 갱신된다.

표 1. DQN 기반의 태스크 오프로딩 코드
Table 1. DQN-based task offloading code

Algorithm 1. DQN based Task offloading

Initialize Q-network Q with random weights
Initialize target Q-network Q' with weights from Q
Initialize experience replay memory D
for episode = 1, M do
  Initialize state s
  for t = 1, T do
    With probability ϵ, select a random action a
    Otherwise, select a = argmax$_a$ Q(s, a)
    Execute action a, observe reward r and next state s'
    Store transition (s, a, r, s') in D
    Sample a random minibatch of transitions from D
    Set y_j = r if the episode ends at step j+1, otherwise set $y_j = r + \gamma$ max_a Q'(s', a)
    Perform a gradient descent step on $(y_j - Q(s, a))^2$ with respect to the network parameters
    Every C steps, update Q' = Q
    s = s'
  end for
end for

표 2. DDPG 기반의 태스크 오프로딩 코드
Table 2. DDPG-based task offloading code

Algorithm 2. DDPG based Task offloading

Initialize actor network π and critic network Q
Initialize target networks π' and Q' with weights from π and Q respectively
Initialize experience replay memory D
for episode = 1, M do
  Initialize state s
  for t = 1, T do
    Select action a = π(s) + noise
    Execute action a, observe reward r and next state s'
    Store transition (s, a, r, s') in D
    Sample a random minibatch of transitions from D
    Set $y_j = r + \gamma$ Q'(s', π'(s'))
    Update the critic by minimizing the loss: L = $(y_j - Q(s, a))^2$
    Update the actor using the sampled policy gradient: $\nabla_{\theta\pi} J \approx E[\nabla_a Q(s, a) | s=s_t, a=\pi(s_t) \nabla_{\theta\pi} \pi(s) | s=s_t]$
    Update the target networks:
    $\theta' = \tau\theta + (1 - \tau)\theta'$
    s = s'
  end for
end for

Ⅳ. 실험 및 성능 비교 분석

제안된 태스크 오프로딩 방법의 성능을 평가하기 위해, 시뮬레이션을 수행하였다. 환경에는 총 3개의 MEC 서버가 배치되어 있으며, 각 서버의 통신 반경은 250m로 가정하였다. 스마트 디바이스에서는 매 타임슬롯 t에서 태스크가 생성되며, 각 타임 슬롯의 길이는 1ms로 설정하였다. 실험 조건으로, 전송 대역폭 B는 10MHz, 잡음 전력 ξ^2는 10^{-9}[17]W로 설정하였다. 또한, 스마트 디바이스의 최대 전송 전력 p_n^{tran}은 0.8W, 최대 CPU 주기 F_n는 1GHz, 비트 당 필요한 CPU 사이클 수 C_n은 500cycle/bit이며, 에너지 계수 μ는 10^{-19}[18]로 설정하였다. 표 3은 이러한 시뮬레이션의 하이퍼 파라미터를 나열한 테이블을 제공한다.

표 3. 실험 하이퍼 파라미터
Table 3. Simulation Hyper-parameters

항목	DQN	DDPG
γ	0.99	0.99
학습률	1e-4	1e-5
리플레이 버퍼 크기	2.5×10^5	2.5×10^5
배치 크기	128	128

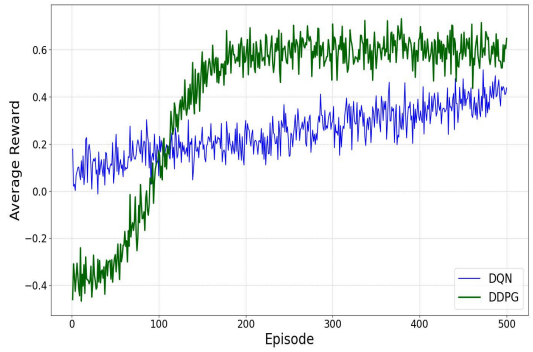

그림 1. DQN과 DDPG의 수렴 분석
Fig. 1. Convergence Analysis of DQN and DDPG

그림 1에서 DQN은 일반적으로 안정적인 수렴을 보이며, 지속적인 학습 과정을 통해 목표값에 접근한다. 초기 단계에서는 보상이 점차 증가하는 경향을 보이나, 계속해서 최적의 해를 찾기 위해 계속해서 학습하는 것을 의미한다. 반면, DDPG는 다양한 환경 변화에 신속하게 대응하는 능력이 있어, 빠른 수렴을 보인다. 특히 연속적

인 행동 공간이 있는 복잡한 환경에서 DDPG의 이런 특성은 큰 이점을 제공한다. DDPG의 빠른 수렴 능력은 높은 보상을 신속하게 획득하는 데 도움을 주어, 문제 해결에 더욱 효율적인 접근을 가능하게 한다.

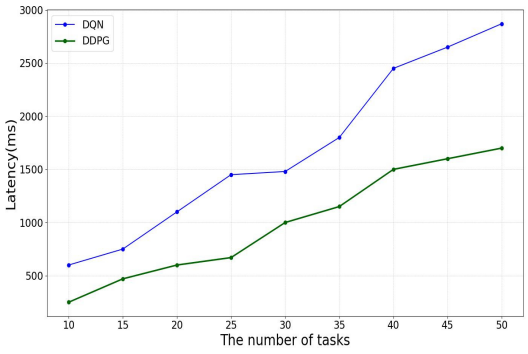

그림 2. 태스크 수에 따른 지연시간
Fig. 2. Impact of the number of tasks on the latency

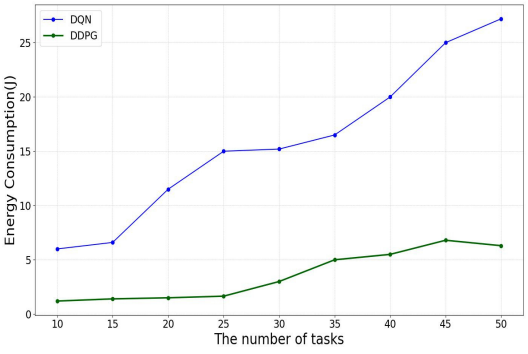

그림 3. 태스크 수에 따른 에너지 소비량
Fig. 3. Impact of the number of tasks on the energy consumption

Fig 2와 3은 태스크 수에 따른 지연시간 및 에너지 소비량을 비교하여 보여준다. 태스크 수가 증가함에 따라 DQN과 DDPG의 지연 시간도 증가하는 경향을 보이는데, 이는 예상된 결과이다. 즉, 처리할 태스크가 많아지면 지연시간도 자연스럽게 증가한다. 그러나 두 알고리즘 간에는 분명한 차이가 있었다. 태스크의 수가 상대적으로 적을 때는 DQN과 DDPG의 사이에 큰 성능 차이가 없었지만, 태스크의 수가 많아질수록 DQN의 지연시간은 DDPG에 비해 더욱 빠르게 증가하는 경향을 보였다. 이러한 결과는 DDPG가 해당 환경에서 DQN보다 더 안정적인 성능을 제공한다는 결론을 도출할 수 있었다. DDPG는 태스크의 수가 증가함에 따라 지연 시간의 증가율이 상대적으로 둔화되는 반면, DQN의 경우 계속 가파르게 증가했다. 그 결과, 많은 태스크 수가 많은 상황에서 DDPG가 더 효율성을 보였다. 이러한 성능 차이의 원인은 알고리즘의 구조, 최적화 전략 등 다양한 요소 때문일 수 있으며, 이에 대한 더 많은 연구와 분석이 필요하다.

V. 결 론

본 연구에서는 커넥티드 홈과 MEC를 통합한 환경에서 의존적인 태스크들의 스케줄링 방안을 제안하였다. 커넥티드 홈의 경우 지연시간 제약 조건이 핵심적인 역할을 하고, 스마트 디바이스는 제한된 배터리 용량 때문에 작업 로드의 효율적인 분산이 요구된다. 이와 같은 문제를 극복하기 위해, 심층 강화학습을 이용하여 태스크의 실행시간을 줄이는 동시에 MEC 서버에서의 로드 분산을 최적화하는 방안을 도출하였다. 실험 결과, 태스크의 수가 증가하더라도 알고리즘이 성능 저하 없이 안정적으로 수행되는 것을 확인할 수 있었다. 앞으로의 연구에서는 MEC의 에너지 효율성을 높이는 것과 작업 할당의 적응성을 중점으로 다룰 계획이다. 또한, 다양한 네트워크 환경과 스마트 디바이스 특성에 따라 스케줄링 전략의 유용성과 효과에 대한 연구도 계획하고 있다.

References

[1] Muhammad Raisul Alam, Mamun Bin Ibne Reaz, Mohd Alauddin Mohd Ali, "A Review of Smart Homes—Past, Present, and Future", IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), Vol. 42, No. 6, pp. 1190-1203, 2012.
DOI: https://doi.org/10.1109/TSMCC.2012.2189204

[2] Y. M. Lee & C. H. Han. (2021). "Research on Touch Function Capable of Real-time Response in Low-end Embedded System." Journal of the Korea Academia-Industrial Cooperation Society, Vol. 22, No. 4, pp. 37-41.
DOI: https://doi.org/10.5762/KAIS.2021.22.4.37

[3] D. Lim, W. Lee, W. T. Kim, I. Joe, "DRL-OS: A Deep Reinforcement Learning-Based Offloading Scheduler in Mobile Edge Computing", Sensors, Vol. 22, No. 23, 9212, 2022.
DOI: https://doi.org/10.3390/s22239212

[4] S. P. Chatrati, G. Hossain, A. Goyal, A. Bhan, S. Bhattacharya, D. Gaurav, S. M. Tiwari, "Smart Home

Health Monitoring System for Predicting Type 2 Diabetes and Hypertension", Journal of King Saud University-Computer and Information Sciences, Vol. 34, No. 3, pp. 862-870, 2022.
DOI: https://doi.org/10.1016/j.jksuci.2020.01.010

[5] Ducsun Lim, Inwhee Joe, "A Delay and Energy-Aware Task Offloading and Resource Optimization in Mobile Edge Computing", In: Computer Science On-line Conference, Cham: Springer International Publishing, pp. 259-268, 2023.
DOI: 10.1007/978-3-031-35317-8_17

[6] Xiang Sun, Nirwan Ansari, "Latency Aware Workload Offloading in the Cloudlet Network", IEEE Communications Letters, Vol. 21, No. 7, pp. 1481-1484, 2017.
DOI: 10.1109/INFCOMW.2014.6849257

[7] Weiwen Zhang et al., "Energy-Optimal Mobile Cloud Computing Under Stochastic Wireless Channel", IEEE Transactions on Wireless Communications, Vol. 12, No. 9, pp. 4569-4581, 2013.
DOI: 10.1109/TWC.2013.072513.121842

[8] R. Kwon & G. Kwon. (2023). "Generating Controller of GR (1) Synthesis and Reinforcement Learning in Game-Solving." Journal of the Korean Society of Information Technology, Vol. 21, No. 7, pp. 13-22.
DOI: 10.14801/jkiit.2023.21.7.13

[9] Matthew Hausknecht, Peter Stone, "Deep Reinforcement Learning in Parameterized Action Space", arXiv preprint arXiv:1511.04143, 2015.
DOI: https://doi.org/10.48550/arXiv.1511.04143

[10] Timothy P. Lillicrap et al., "Continuous Control with Deep Reinforcement Learning", arXiv preprint arXiv:1509.02971, 2015.
DOI: https://doi.org/10.48550/arXiv.1509.02971

[11] C. Qiu, Y. Hu, Y. Chen, B. Zeng, "Deep Deterministic Policy Gradient (DDPG)-Based Energy Harvesting Wireless Communications", IEEE Internet of Things Journal, Vol. 6, No. 5, pp. 8577-8588, 2019.
DOI: 10.1109/JIOT.2019.2921159

[12] Robert B. Cooper, "Queueing Theory", In: Proceedings of the ACM'81 Conference, pp. 119-122, 1981.
DOI: https://doi.org/10.1145/800175.809851

[13] Scott Fujimoto, David Meger, Doina Precup, "Off-policy Deep Reinforcement Learning without Exploration", In: International Conference on Machine Learning, PMLR, pp. 2052-2062, 2019.
DOI: https://doi.org/10.48550/arXiv.1812.02900

[14] Seo-Yeon Gu, Seok-Jae Moon, Byung-Joon Park, "Reinforcement Learning Multi-Agent Using Unsupervised Learning in a Distributed Cloud Environment", International Journal of Internet, Broadcasting and Communication, Vol. 14, No. 2, pp. 192-198, 2022.
DOI: https://doi.org/10.7236/IJIBC.2022.14.2.192

[15] J. Park, "Self-Awareness and Coping Behavior of Smartphone Dependence among Undergraduate Students", Journal of the Korea Academia-Industrial Cooperation Society (JKAIS), Vol. 22, No. 2, pp. 336-344, 2021.
DOI: https://doi.org/10.5762/KAIS.2021.22.2.336

[16] Eun-Gyu Ham, Chang-Bok Kim, "Model Implementation of Reinforcement Learning for Trading Prediction Using Deep Q Network", The Journal of KIIT, Vol. 17, No. 4, pp. 1-8, 2019..
DOI: 10.14801/jkiit.2019.17.4.1

[17] Duc-Sun Lim, Yeon-Ah Min, & Dong-Kyun Lim. (2023). "Deep Reinforcement Learning Based University Major Recommendation System." Journal of the Korean Internet Broadcasting and Telecommunication Society, Vol. 23, No. 4, pp. 9-15.
DOI: https://doi.org/10.7236/JIIBC.2023.23.4.9

[18] Ducsun Lim, Inwhee Joe, "A DRL-Based Task Offloading Scheme for Server Decision-Making in Multi-Access Edge Computing", Electronics, Vol. 12, No. 18, 3882, 2023.
DOI: https://doi.org/10.3390/electronics12183882

저 자 소 개

임 덕 선(정회원)

- 2004년 2월 : 한양사이버대학교 컴퓨터공학과 학사
- 2020년 : 한양사이버대학교 컴퓨터공학과 교수
- 2023년 8월 : 한양대학교 일반대학원 컴퓨터·소프트웨어학과 석·박사
- 관심분야 : 강화학습, 모바일 엣지 컴퓨팅, Anomaly Detection, 6G

손 규 식(정회원)

- 1982년 2월 : 한양대학교 전자과 졸업 (학사)
- 1984년 2월 : 한양대학교 대학원 전자통신공학과 졸업 (석사)
- 2003년 8월 : KAIST 전기전자공학과 졸업 박사
- 2004년 3월 : 한양사이버대학교 해킹보안학과 교수
- 관심분야 : 정보보안, 사물인터넷, 인터넷 신뢰성