논문 21-46-12-15 The Journal of Korean Institute of Communications and Information Sciences '21-12 Vol.46 No.12
https://doi.org/10.7840/kics.2021.46.12.2212

에너지 수집형 IoT 엣지 컴퓨팅 환경에서 사용자 QoE를 높이기 위한 Q-러닝 기반 동적 태스크 오프로딩 기법

강 민 재◆, 이 승 우*, 공 유 진*, 김 영 현**, 윤 익 준***, 노 동 건°

Q-Learning Based Dynamic Task-Offloading Scheme to Improve QoE in Energy Harvesting IoT Edge Computing Environments

Minjae Kang◆, Seungwoo Lee*, Yujin Gong*, Younghyun Kim**,
Ikjune Yoon***, Dong Kun Noh°

요 약

일반적인 IoT 장치는 배터리를 에너지원으로 사용하기 때문에 제한적인 수명을 가질 수밖에 없다. 따라서 이 문제를 근본적으로 해결하기 위해 최근에는 태양에너지를 중심으로 다양한 환경 에너지 수집형 IoT에 관한 연구가 진행되고 있다. 한편, IoT 장치의 에너지 문제와 지연시간과 같은 측면에서 QoE를 향상시키고자 많은 연구가 진행되고 있는데, 이러한 QoE 향상을 위한 대부분의 연구들은 비용 최소화 문제, 특별히 에너지 사용 최소화 문제로 접근하고 있다. 그러나 제한적인 에너지로 인한 이러한 에너지 비용 최소화 방향의 접근은 지속적인 에너지 수집이 가능한 에너지 수집형 IoT 장치에는 적합하지 않다. 많은 에너지 수집형 IoT 장치들은 지속적으로 에너지를 수집할 수 있기 때문에, 비용 최소화적 접근 보다는 수집되는 에너지를 최대한 활용할 수 있도록 하는 비용 최적화 방향으로 접근해야 한다. 본 논문에서는 에너지 수집형 IoT 엣지 컴퓨팅 환경에서, 수집에너지 활용을 최대화 하면서 사용자의 QoE를 향상시킬 수 있는, 강화학습 기반의 동적 태스크 오프로딩 기법을 제안한다. 제안된 기법은, 동적으로 변하는 에너지 수집량과 IoT 장치 및 엣지 노드의 정보를 이용해 Q-learning을 모델링 했으며, 이는 기존 최소화 문제의 복잡한 모델링과 비교해 매우 단순하지만, 효율적인 동작을 할 수 있다. 결과적으로 제안 기법은 EH-IoT 응용 사용자에게, 수집된 에너지를 최대한 활용해 안정적으로 오랜 기간 사용할 수 있는 높은 안정성을 제공하고, 지연시간 측면에서도 빠른 응답시간을 보여줌으로써, 높은 QoE를 제공할 수 있다.

Key Words : IoT, Mobile Edge Computing, Task-offloading, Reinforcement Learning, Q-learning Internet of things, LoraWAN, Solar-powered, Wireless power transmission, Simulation

ABSTRACT

Typical IoT(Internet of things) devices have a limited lifetime because they use a battery as an energy source. To fundamentally solve this problem, various environmental energy harvesting IoT is being researched around solar energy. Meanwhile, many studies are attempting to improve the quality of experience (QoE) in

※ 이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. NRF- 2021R1A2C1005919).
◆ First Author : Soongsil University Foundation of Univ.-Industry Coop., minjaekang@ssu.ac.kr, 정회원
° Corresponding Author : Soongsil University School of AI Convergence, dnoh@ssu.ac.kr, 종신회원
* Soongsil University Department of of Intelligent System, {klsw725, 1102177006}@soongsil.ac.kr, 학생회원
** University of Wisconsin, Madison Department of Elec. and Comp. Engineering, younghyun.kim@wisc.edu
*** Kyonggi University School of AI Computer Engineering, ijyoon@kyonggi.ac.kr, 정회원
논문번호 : 202107-152-B-RN, Received July 2, 2021; Revised August 31, 2021; Accepted September 8, 2021

terms of the energy problem and delay time of IoT devices. Many of these studies on QoE improvement are approaching it as a cost minimization problem, especially an energy use minimization problem. However, such an energy cost minimization approach due to limited energy is not appropriate for energy harvesting IoT devices, which can continuously harvest energy. Since many energy-harvesting IoT devices can continuously harvest energy, we should approach in the direction of cost optimization that can utilize energy to the maximum rather than cost minimization approach. This study proposes a reinforcement learning-based dynamic task offloading scheme that can enhance user QoE while maximizing the utilization of harvested energy in an energy harvesting IoT edge computing environment. The proposed scheme modeled Q-learning using the information of dynamically changing amount of energy harvesting and the information of IoT devices and edge nodes. It can perform very simple, yet efficiently operates compared to complex models of the conventional minimization problem.

Ⅰ. Introduction

센서, 카메라 및 웨어러블 장치와 같은 사물인터넷(IoT) 장치는 제한된 컴퓨팅과 에너지, 메모리 자원을 가지고 있다. 이로 인해 얼굴 인식 응용과 같은 고도화된 응용에서 프로세싱 병목 현상이 발생한다[1]. 이러한 고도화된 IoT 응용에서 사용자가 직접적으로 느끼는 중요한 성능은 IoT 기기의 에너지 효율성(가용시간, 안정적 동작)과 응답 지연시간(태스크 완료시간), 그리고 응답 정확성(결과 정확도, 만족도)이다. 본 연구에서는 이를 IoT 응용의 QoE(quality of experience, 사용자 체감 품질)라 정의한다. 이러한 문제는 모바일 엣지 컴퓨팅(MEC) 기술[2]로 해결할 수 있다. 이는 기존 IoT 장치에서 수행하던 태스크를 기지국, 엑세스 포인트와 같은 엣지 노드로 태스크를 오프로딩하여 해결하며, 엣지 노드의 컴퓨팅과 에너지, 메모리 자원을 사용함으로 프로세싱의 지연을 줄여주고, 에너지를 절약하여 결과적으로 사용자의 QoE를 증가시켜 만족도를 높일 수 있다.

태스크 오프로딩 기법에서 해결해야 할 이슈는 다음과 같다. 연결된 후보 엣지 노드 중 타겟 엣지 노드를 선택하고[3], 오프로딩의 비율 결정, 즉 엣지 노드로 오프로드 할 태스크의 크기를 결정해야 하며[4][5], 무선 채널의 페이딩이나 간섭으로 인해 다양하게 변화하는 무선 링크의 전송속도[6]를 고려하는 것이다. 따라서 한정적인 에너지와 제한된 시간 안에 최적의 오프로딩 정책을 찾는 것은 많은 어려움이 있다. 최근 오프로딩에 여러 가지 방법을 적용해 최적의 정책을 찾는 연구가 많이 진행되고 있다. 예를 들어 [7]의 연구에서 제안한 모바일 오프로딩 기법은 Lyapunov 최적화를 사용해 전송 지연 모델과 로컬 실행 모델에 대한 지식을 기반으로 단일 MEC 서버로 오프로딩 할 때의 컴퓨팅 비용을 계산하였다. [8]의 연구에서 제안한 기법은 bi-objective 최적화 문제를 해결하기 위해 휴리스틱 알고리즘을 제시하였고, [9]의 연구에서 제안한 기법은 오프로딩의 여부를 결정하기 위해 그리디 휴리스틱 알고리즘을 이용해 계산하였다. 이러한 접근방식의 연구들은 엣지 노드가 여러 개 있거나, 동적 MEC 네트워크와 같이 계속 변화하는 환경에서 실시간으로 최적의 정책을 결정하지 못한다는 한계가 있다.

한편 에너지 수집(EH) 기술은 배터리의 수명을 반영구적으로 늘려주어 IoT 장치의 QoE를 증가시켜 주는 유망한 기술이다. 일반적인 IoT 장치는 배터리로 동작되기 때문에 노드의 수명에 제한이 있고, 사람이 접근하기 힘든 지역에서 배터리 교체와 같은 유지보수가 어렵다는 문제가 있다. 이러한 문제점들을 해결하기 위해 EH-IoT를 이용한 많은 연구가 진행되었다[10]. EH은 일반적으로 주변 환경 에너지(태양, 풍력, 무선 전력 전송, 압력 등)를 에너지원으로 사용하며, 각각의 에너지원에 따라 다양한 특성들이 있다. 그중 태양에너지는 에너지 밀도, 주기적인 수집, 높은 예측성으로 IoT 장치에서 많이 사용된다.

본 논문에서는 동적 MEC 네트워크에서 반영구적으로 동작하는 EH-IoT를 위한 강화학습(RL) 기반 동적 태스크 오프로딩 기법을 제안한다. 이는 IoT 장치와 엣지 노드 사이의 무선 전송속도와 예측된 에너지, IoT 장치의 배터리 레벨에 따라 최적화된 정책을 결정하며, 이를 Markov decision process(MDP)로 모델링할 수 있다. 따라서 IoT 장치는 Q-learning과 같은 RL 기술의 학습 에이전트(learning agent) 역할을 수행할 수 있고, 지속적으로 오프로딩 작업을 거친 후 최적의 오프로딩 정책을 찾을 수 있다.

구체적으로 현재 IoT 장치의 상태를 상태(state)로 정의하고, 오프로딩할 타겟 엣지 노드의 선택과 태스

크의 비율을 액션(action)으로 정의한다. 오프로딩 정책은 상태-액션 쌍을 이용해, 현재 상태와 Q-함수에 따라 결정된다. 그 후 IoT 장치는 각 시간 슬롯에서 오프로딩 동작에 따른 전체 지연시간, 에너지 활용도, 태스크 미수행 손실을 기반으로 보상을 평가하며, Bellman equation[11]에 따라 보상을 반복적으로 학습에 반영해 Q-함수가 업데이트된다. 또한 오프로딩 프로세스 초기단계에서 소모되는 탐색시간을 단축할 수 있도록 RL의 전이 학습 기법(TL)[12]을 사용한다. TL은 유사한 시나리오에서 오프로딩 경험을 활용하여 보상을 초기화하기 때문에 RL의 학습 속도를 증가시킬 수 있다. 본 연구의 차별적인 기여는 다음과 같다.

- EH-IoT를 위한 RL기반 태스크 오프로딩 기법을 제안한다. 이를 통해 IoT 장치는 현재 배터리 레벨, 무선 전송속도와 예측된 에너지 수집 모델에 따라 엣지 노드와 오프로딩 비율을 선택한다.
- TL을 사용하여 초기 단계의 학습 속도를 가속화하여 오프로딩 성능을 향상한다.
- 따라서 에너지 수집형 엣지컴퓨팅 환경에서 사용자의 QoE(응답지연시간, 에너지 활용도)를 향상한다.

본 논문의 구성은 다음과 같다. II장에서 시스템 모델에 대하여 설명하고, III장에서는 제안 기법을 자세히 기술한다. IV장에서는 실험을 통하여 제안 기법을 검증하고, 마지막으로 V장에서 결론으로 마무리한다.

Ⅱ. System Model

본 절에서는 제안하는 기법의 시스템 개요와 대략적인 동작을 설명하고 RL을 수행하기 위해 에너지 관점에서 시스템을 모델링 한다. Fig. 1.은 IoT 장치와 M개의 엣지 노드로 구성된 MEC 시스템의 개요를 보여주고 있다. IoT 장치는 스마트 시계와 스마트폰 등의 장치를 의미하며, 태양광 패널과 같은 에너지 수집 모듈과 배터리가 장착되어 있다. 이러한 IoT 장치는 보통 센싱된 데이터를 로컬에서 처리한다. 하지만 처리해야 되는 태스크가 얼굴 인식 응용과 같은 고도화된 작업이라면 로컬에서 처리하는데 지연시간과 에너지 측면에서 제한적이다. 따라서 얼굴 인식 응용과 같은 연산 집약적 태스크는 엣지 노드로 처리해야 할 태스크를 전송하고 처리된 결과를 받음으로써, IoT 장치의 에너지 및 지연시간에 도움이 될 수 있다. 본 논문에서는 이렇게 엣지 노드로 태스크를 전달하여 처리하는 것을 태스크 오프로딩이라 정의한다.

제안하는 기법은 다음과 같은 순서로 동작하며, Fig. 1.과 같다. 일정한 간격으로 시간을 나누고 이는 $k\,(k\in 1,2,\ldots)$로 정한다. IoT 장치는 시간 k에서 C^k비트만큼 처리해야 할 데이터가 센싱된다. 그 후 [13]에서 제안한 Computation Partition Scheme을 이용해 태스크를 N_x 단위로 나눈다. IoT 장치는 무선 전송속도 $B_i^{(k)}\,(1 \le i \le M)$를 기반으로 시간 k에서 태스크를 오프로딩 할 엣지 노드i를 선택하고, 오프로딩 비율 $x^{(k)}(x^{(k)} \in \{l/N_x\}_{0 \le l \le N_x})$를 결정한다. 구체적으로 IoT 장치는 $x^{(k)}=0$ 일 때 로컬에서 모든 태스크를 처리하고, $x^{(k)}=1$ 일 때 모든 태스크를 오프로딩 한다. 따라서 IoT 장치는 $x^{(k)}C^{(k)}$ 비트를 타겟 에지 노드 i로 오프로딩 하고, 나머지 $(1-x^{(k)})C^{(k)}$비트는 로컬에서 처리한다 $(0 < x^{(k)} < 1)$. IoT 장치는 시간 k에서 오프로딩 정책 $\mathbf{a}^{(k)}=[i^{(k)},x^{(k)}]\,(\mathbf{a}\in\mathbf{A})$을 선택한다. 각각의 기호는 Table 1.에서 정리한다. 또한 시간 표기가 의미가 없는 경우 시간 인덱스 k는 생략한다.

제안하는 기법은, 태스크를 로컬에서 수행 시 소모되는 에너지와 지연시간을 엣지 노드로의 오프로딩 시 소모되는 양과 비교함으로써, 각 태스크의 오프로딩 여부 및 타겟 엣지 노드를 선택하고 있다. 결과적으로 IoT 장치의 태스크 처리 지연시간과 정전시간을 감소시켜 사용자의 QoE를 증가시킬 수 있게 된다. II.1. 절에서는 태양에너지 수집 및 할당 모델을 설명하고 있는데, 구체적으로 노드가 각 시간 슬롯에 에너지를 얼마나 할당할지에 관한 모델링이며, 이는 에너지가 수집되지 않는 밤 시간대에도 안정적인 동작 수행을 가능케 하도록 설계되었다. 이렇게 각 노드의 타임슬롯 별로 할당된 에너지양을 기반으로, II.2. 절과

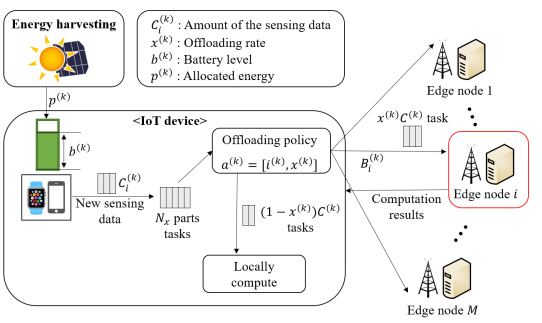


그림 1. 제안 기법의 개요
Fig. 1. Overview of the Proposed Scheme

표 1. 주요 심볼
Table 1. Table of Major Symbols

Symbol	Description
$x^{(k)}$	Ratio of offloading to edge node
$B_i^{(k)}$	Wireless transmission speed between IoT device and edge node
$p^{(k)}$	Allocated energy
$b^{(k)}$	Battery level
f	CPU frequency (frequency)
N	CPU cycle (cycle)
$E^{(k)}$	Consumed energy
$T^{(k)}$	Delay time
ψ	Weight of unperformed task

II.3. 절에서는 각각, 로컬에서 태스크를 처리할 때 지연시간과 에너지 소모양/잔여 에너지양 예측 방법, 태스크 오프로딩시 지연시간과 에너지 소모양/잔여 에너지양 예측 방법에 대해 설명하고 있다.

2.1 Energy Harvesting Model

IoT 장치는 주변 환경(태양, 바람, 무선 전력 전송, 압력)에서 에너지를 수집할 수 있으며, 그중 태양에너지는 높은 에너지 밀도를 가지고 있고, 주기적으로 수집되며, 정확한 예측이 가능하여 많은 분야에서 사용되고 있다.

따라서 본 연구에서는 에너지 수집 자원으로 태양에너지를 사용한다. 주의해야 할 것으로 태양에너지는 일몰 이후 수집되는 에너지가 거의 없어 노드의 정전을 유발할 수 있으며, 낮에는 에너지 수집양이 많아 저장 가능한 배터리 용량을 넘어서는 잉여 에너지가 발생할 수 있다. 따라서 시간과 무관하게 균일한 에너지 소비를 위해 균형적으로 에너지를 할당하는 방식이 필요하다. [15]는 이러한 점들을 고려한 균형적인 에너지 할당 방식을 제안하며, Fig. 2.와 같이 수집에너지를 각 단위 시간 슬롯에 효율적이고 균등하게 할당할 수 있다. 아래는 [15]에서 제안한 에너지 할당 기법이다.

$$p^{(k)} = E_{\text{alloc}}^{(k)} - E_{\text{sys}}^{(k)} \tag{1}$$

$p^{(k)}$는 시간 슬롯 k에서 태스크 처리를 위해 할당된 에너지양이며, $E_{\text{alloc}}^{(k)}$는 시간 슬롯 k 에 할당된

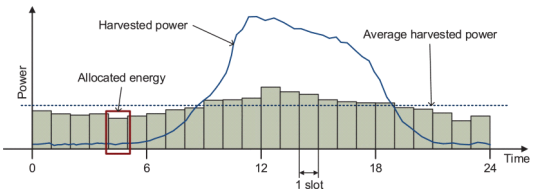


그림 2. 에너지 할당 기법
Fig. 2. Energy Allocation Technique[14]

총 수집에너지이며, $E_{\text{sys}}^{(k)}$는 태스크 처리에 사용되는 에너지를 제외한 모든 소비 에너지이다.

2.2 Latency and Energy Model on Local Computing

IoT 장치는 로컬에서 CPU를 사용하여 태스크를 처리한다. IoT 장치는 각 시간 슬롯 k에서 $(1-x^{(k)})C^{(k)}$비트만큼 태스크를 처리해야 한다. CPU의 성능은 사이클(cycle)과 주파수(frequency)에 따라 다르며, N은 1비트를 처리하는데 필요한 CPU의 사이클 수를 나타낸다. 따라서 $(1-x^{(k)})C^{(k)}$비트에 필요한 총 CPU 사이클 수는 $(1-x^{(k)})C^{(k)}N$이 된다.

또한, 에너지 소비를 제어하기 위해 동적 전압 및 주파수 스케일링 기술[16]을 사용할 수 있는데, IoT 장치는 사이클 m $(m \in \{1, 2, ..., (1-x^{(k)})C^{(k)}N\})$에서 주파수 f_m을 조절해 에너지 소비를 제어한다. 시간 슬롯 k에서 로컬 컴퓨팅 실행 대기 시간은 $T_0^{(k)}$로 표기하고, 아래와 같이 정의한다.

$$T_0^{(k)} = \sum_{m=1}^{(1-x^{(k)})C^{(k)}N} \frac{1}{f_m} \tag{2}$$

[17]의 연구에 따르면 IoT 장치는 시간 k에서 $E_0^{(k)}$에너지를 소비하며 아래와 같다.

$$E_0^{(k)} = \sum_{m=1}^{(1-x^{(k)})C^{(k)}N} \varsigma f_m^2, \tag{3}$$

여기서 ς는 칩 아키텍처에 따른 유효한 커피시턴스 계수 이다. 제안된 기법은 [16]와 [17]에 따른 동적 주파수 및 전압 스케일링을 가진 IoT 장치에 적용될 수 있다.

2.3 Latency and Energy Model on Task Offloading

IoT 장치는 태스크를 오프로딩 할 때 무선 전송속도 $B_i^{(k)}$에 따라 전송 시간이 달라지며, 엣지 노드i의 CPU에 따라 태스크의 처리 속도가 달라진다. 시간 슬롯 k에서 엣지 노드 i로 태스크를 오프로딩 할 때, $x^{(k)}C^{(k)}$비트를 처리하는 시간은 $T_i^{(k)}$로 표기하고, 아래와 같다.

$$T_i^{(k)} = \frac{x^{(k)}C^{(k)}}{B_i^{(k)}} + \sum_{m_i=1}^{(1-x^{(k)})C^{(k)}N} \frac{1}{f_{m_i}} \tag{4}$$

[15]에 따라 IoT 장치는 태스크를 오프로딩할 때 에너지 $E_i^{(k)}$를 소비한다. 이는 전송 전력 P와 전송 시간에 따라 다르며, 아래와 같다.

$$E_i^{(k)} = \frac{x^{(k)}C^{(k)}P}{B_i^{(k)}} \tag{5}$$

시간 k에서 IoT 장치가 오프로딩을 포함하여 태스크를 위해 소비하는 총 에너지는 $E^{(k)} = E_0^{(k)} + E_i^{(k)}$로 정의한다. 시간 k에서 배터리 레벨은 $b^{(k)}$로 정의하고, 아래와 같이 계산한다.

$$b^{(k+1)} = \max\{0, b^{(k)} - E^{(k)} + p^{(k)}\} \tag{6}$$

각 절의 모델을 기반으로 본 논문에서는 태스크의 오프로딩 비율과 타겟 엣지 노드를 선택하는 기법을 제안한다. 제안하는 기법은 기존의 최적화 방법을 이용한 기법이 아닌 동적인 환경에 대응하기 적합한 Q-learning을 이용해 최적의 오프로딩 비율과 타겟 엣지 노드를 선택한다.

Ⅲ. Q-learning-based Task Offloading Scheme

본 절에서는 제안하는 기법인 Q-learning 기반 태스크 오프로딩 기법(QTO)을 설명하며, 요약하면 다음과 같다. ①태스크 오프로딩을 위한 MDP(Markov Decision Processes)에 관해 설명하고, ②매개 변수의 초기값 설정에 관해 설명하며, ③상태와 액션을 정의하고 ④실행된 결과를 받아 평가하고, ⑤Q-함수를 업데이트한다. 제안하는 태스크 오프로딩 기법은 [18]에서 제안한 기법을 기반으로, 이를 태양에너지 수집 모델 및 에너지 할당 기법에 최적화하여 수정 설계되었다.

IoT 장치에서 엣지 노드로 오프로딩하는 것은 현재의 상태와 액션이기 때문에 MDP로 정의가 가능하다. 구체적으로 태스크 오프로딩을 수행할 때, IoT 장치는 이전 시간$(k-1)$의 무선 전송속도와 할당된 에너지, 현재 배터리 레벨을 기반으로 엣지 노드의 선택과 오프로딩 비율을 결정한다. 주목할 점은 IoT 장치가 태스크 오프로딩을 위해 필요한 정보는 과거 기록이 아니라 현재 상태와 액션이라는 점이다. 따라서 태스크 오프로딩 프로세스는 MDP로 정의가 가능하며, 널리 사용되는 RL 기법인 Q-learning 기법을 적용해 최적의 정책을 구할 수 있다. 이는 일반적으로 사용하는 최적화 기법에서 필요한 MEC 모델, 에너지 소비 모델 및 컴퓨팅 지연 모델이 없어도 된다는 장점이 있다.

RL의 초기 단계는 학습이 많이 이루어지지 않았기 때문에 Q-value가 좋지 않다. 따라서 탐색(exploration)의 비중이 높기 때문에 이를 보완하기 위해 매개 변수의 초기값을 잘 설정해야 한다. 제안하는 QTO는 학습 매개 변수를 초기화하기 위해 TL[12]을 사용한다. 구체적으로, 이는 유사한 환경의 오프로딩 경험을 이용해 Q-value를 초기화한다는 것이다. 이로 인해 태스크 오프로딩 프로세스의 초기 단계에서 수행될 무작위 탐색을 줄여주게 되고, 이는 학습 속도를 가속화 한다.

Q-learning을 하기 위해 상태-액션 쌍이 필요하며, 상태는 할당된 에너지양, 배터리 레벨, 무선 전송속도로 정의된다. 구체적으로 IoT 장치는 수식(5)에 따라 할당된 에너지양 $p^{(k)}$를 예측하고, 현재 배터리 레벨 $b^{(k)}$와 엣지 노드 M의 이전 무선 전송속도 $B_1^{(k-1)}, ..., B_M^{(k-1)}$를 상태로 정의하고, $s^{(k)} = [B_1^{k-1}, ..., B_M^{(k-1)}, p^{(k)}, b^{(k)}]$로 표현할 수 있다.

액션은 엣지 노드 i와 오프로딩 비율로 정의하며, 다음과 같이 표기한다 $a^{(k)} = [i^{(k)}, x^{(k)}]$. 또한 액션의 선택은 Q-value를 최적화하기 위한 탐색(exploration)과 현재의 Q-value에 따라 효율적으로 동작하는 이용(exploitation) 두 가지를 선택할 수 있다. 이 두 가지는 트레이드 오프 관계에 있으며 구체적으로, 탐색은 최적화된 Q-value로 동작하는 것이 아니라 무작위로 동작해 Q-함수를 업데이트하는 것이 목적이다. 반면 이용은 현재의 Q-value가 최적이라고

생각하고 그대로 동작하는 것이다. 제안 기법은 decaying ε-greedy 정책을 적용해 초반에는 탐색(exploration)의 비율을 크게하고 Q-learning이 반복될수록 이용(exploitation)의 비율을 늘리는 방법을 사용한다.

선택된 액션을 수행한 후 IoT 장치는 태스크의 실행 결과를 얻을 수 있다. 구체적으로 IoT 장치는 처리해야 할 전체 태스크 중 $x^{(k)}C^{(k)}$비트를 에지 노드 i로 오프로딩하고, $(1-x^{(k)})C^{(k)}$비트는 로컬에서 컴퓨팅한다. 그 후 엣지 노드는 태스크를 완료하고 결과를 IoT 장치로 보낸다. 이때 완료된 결과 데이터 크기는 작기 때문에 엣지 노드와 IoT 장치 간의 전송 지연은 없다고 가정한다.

액션을 수행 후 결과를 받고 나면 액션의 평가를 해야 한다. 평가 항목은 전체 지연시간, 에너지 활용도, 태스크 미수행 손실, 태스크 오프로딩 이득으로 정의된다. 이때 전체 지연시간은 $T^{(k)}=\max\left\{T_0^{(k)}, T_i^{(k)}\right\}$로 정의하며, IoT 장치의 에너지가 부족한 경우에는 태스크를 수행하지 못하기 때문에 태스크 미수행에 대한 평가를 해야한다. 태스크 미수행 비용은 ψ로 정의하며, $\mathbf{I}(\varpi)$는 ϖ가 참이면 1, 거짓이면 0으로 정의한다. β는 에너지 소비의 가중치, μ는 지연시간의 가중치이다. IoT 장치의 보상은 $R_i^{(k)}(x)$로 표시되며, 이 값은 전체 지연시간, 에너지 활용도, 태스크 미수행 손실, 태스크 오프로딩 이득에 달라지며 아래와 같다.

$$R_i^{(k)}(x)=xC^{(k)}-\psi\mathbf{I}(b^{(k+1)}=0)-\beta E^{(k)}-\mu T^{(k)} \quad (7)$$

평가가 완료된 후 평가를 기반으로 Q-함수를 업데이트해서 정책의 질을 높일 수 있다. 이를 반복할수록 최적의 정책을 찾을 수 있다. 구체적으로 IoT 장치가 상태 $\mathrm{s}^{(k)}$에서 $x^{(k)}C^{(k)}$비트를 엣지 노드i로 오프로딩하면 상태는 $\mathrm{s}^{(k+1)}$로 전환된다. 이를 기반으로, 지속적인 오프로딩 경험 $(\mathrm{s}^{(k)}, \mathbf{a}^{(k)}, R^{(k)}, \mathrm{s}^{(k+1)})$에 의해 IoT 장치는 Q-함수를 업데이트하며 아래와 같이 표현할 수 있다.

$$Q(\mathrm{s}^k, \mathbf{a}^{(k)}) \leftarrow (1-\alpha)Q(\mathrm{s}^k, \mathbf{a}^{(k)}) + \alpha\left(R^k+\gamma\max_{\mathbf{a}'\in\mathbf{A}}Q(\mathrm{s}^{k+1}, \mathbf{a}')\right) \quad (8)$$

여기서 학습 비율 α는 현재 오프로딩의 가중치이며, γ는 미래 보상의 감소 계수이다.

Ⅳ. Simulation Result

이번장에서는 제안 기법의 성능 검증을 위해 시뮬레이션한 결과에 대해 설명한다. 시뮬레이션의 토폴로지는 엣지 노드 3개와 IoT 장치 5, 10, 15개로 구성된 동적 네트워크이며, duration은 시뮬레이션 1회 당 3일 하였고 여러번 반복하여 평균을 구하였다. 에너지 수집 모델은 [14]에서 제안한 에너지 할당 기법을 사용하였다. 또한 제안 기법의 파라미터인 ψ, β, μ, α, γ, ϵ은 각각 10, 0.7, 1, 0.9, 0.5, 0.1로 설정하였다.

시뮬레이션은 제안 기법인 QTO와 에너지 수집에 대한 고려가 없는 일반적인 Q-learning 기반 기법인 Q[17]와 non-offloading scheme, 휴리스틱 알고리즘을 사용해 태스크를 오프로딩하는 DACO[18], HGOS[19]를 포함하여 총 5가지를 비교군으로 IoT 장치의 정전시간, 지연시간, 태스크 미수행 측면에서 진행하였다.

Fig. 3. ~ Fig. 5.에서 볼 수 있듯 10개의 IoT 장치의 경우 Non-offloading scheme과 비교해 Q[17]의 지연시간은 54%, 태스크 미수행은 11% 감소한다. 이것은 수집에너지가 고려되지 않는 것을 포함해, Q-learning의 보상 설정이 적절하게 되지 않았음에도 불구하고 태스크 오프로딩을 하지 않는 것보다는 좋은 결과를 보여주고 있다. 또한 Q[17]와 비교해 제안 기법인 QTO는 전체 시뮬레이션 타임 동안 정전시간은 69%, 지연시간은 10%, 태스크 미수행은 66% 감소한다. 이것은 같은 Q-learning 기반의 기법이라 하더라도 알고리즘의 설계에 따라 성능의 차이가 발생하는 것을 볼 수 있다. 또한 에너지를 고려하지만 휴리스틱 알고리즘인 DACO, HGOS와 제안 기법을 비교해보면, 각각 정전시간은 40%, 65%, 지연시간은 17%, 39%, 태스크 미수행은 39%, 60% 감소한다. 이

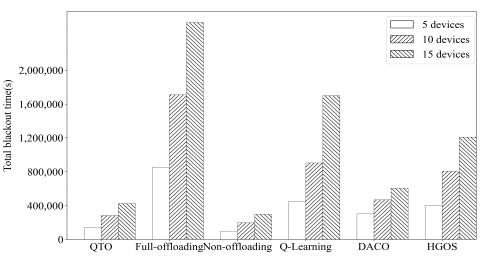


그림 3. IoT 장치 개수에 따른 전체 정전 시간
Fig. 3. Total Blackout time according to the number of IoT devices

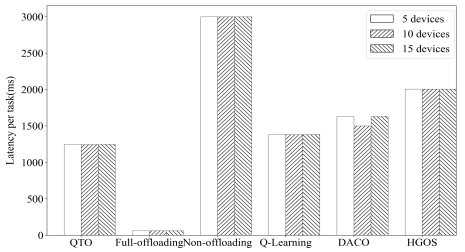


그림 4. IoT 장치 개수에 따른 태스크 당 지연시간
Fig. 4. Latency per task according to the number of IoT devices

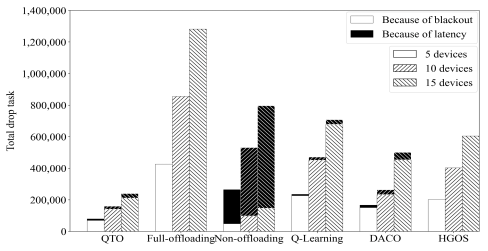


그림 5. IoT 장치 개수에 따른 전체 미수행 태스크
Fig. 5. Total drop task according to the number of IoT devices

는 동적 MEC 네트워크와 같이 계속 변화하는 환경에서 실시간으로 최적의 정책을 결정하지 못한다는 한계와 결정을 위한 오버헤드에 따라 성능의 차이가 발생하는 것을 볼 수 있다. IoT 장치의 개수에 따른 결과도 10개의 경우와 큰 차이가 없이 제안 기법이 우수한 것을 볼 수 있다.

V. Conclusions and Future Work

본 논문은 태양에너지 기반 IoT 장치를 사용하는 환경에서 사용자의 QoE(응답시간, 에너지 활용도, 정확도)를 높이기 위해 강화학습 중 많이 사용되는 Q-learning을 이용해 동적 태스크 오프로딩 기법을 제안하고 있다. 기존 많이 사용되는 Lyapunov 최적화 또는 convex 최적화 기법은 등은 비용(즉, 에너지 사용량 또는 작업 지연시간) 최소화 문제로 접근하고, 이러한 접근방식은 IoT 전체 상황에 대한 모든 정보가 필요하기 때문에 주변 노드나 링크 상태, 에너지 상태가 계속해서 변화하는 환경에서 동적으로 결정을 내리지 못한다는 한계가 있었다. 제안 기법은 이러한 한계점이 없는 강화학습(현재 시점의 해당 노드의 정보만 필요함)을 이용해 태스크 오프로딩 기법을 제안하였으며, 이는 작업이 반복될수록 최적의 동작을 하게 된다. 또한 최적화 기법과는 다르게 구체적인 시스템 모델이 필요하지 않기 때문에 다양한 응용에 쉽게 적용될 수 있다. 이렇게 강화학습 기반의 단순하지만, 효율적인 오프로딩 기법을 사용함으로써 EH-IoT 응용 사용자에게 에너지 QoE 측면에서는 수집에너지를 최대한 사용하며 안정적으로 오랜 기간 사용할 수 있는 기회를 제공하고, 응답 지연 시간 QoE 측면에서도 빠른 응답시간을 기대할 수 있다.

향후 연구 과제로는 첫 번째, 태스크 오프로딩을 기존 Q-learning이 아닌 Deep Q-Networks(DQN)로 구현하는 것이다. Q-learning의 학습 속도는 상태-액션 공간의 차원에 따라 달라진다. 즉, 차원을 늘릴수록 에너지 사용량은 많아지고, 그에 대응하는 채널 상태도 필요하다. 이는 Deep Neural Network(DNN)에서 신경망을 Q-함수로 사용하는 DQN을 이용해 태스크 오프로딩을 구현하면 해결할 수 있을 것으로 예상된다. 두 번째로는 bias에 대한 고려가 필요하다. Q-learning은 TD 방식이기 때문에 variance는 작지만, bias가 발생할 확률이 높다. 따라서 MC 방식과 비교하여 bias의 발생 확률을 검증해 해결할 수 있을 것으로 예상된다.

References

[1] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surv. Tut.*, vol. 19, no. 4, pp. 2322-2358, Aug. 2017.

[2] J. Feng, Z. Liu, C. Wu, and Y. Ji, "AVE: Autonomous vehicular edge computing framework with ACO-based scheduling," *IEEE Trans. Veh. Technol.*, vol. 66, no. 12, pp. 10660-10675, Dec. 2017.

[3] T. Q. Dinh, J. Tang, Q. D. La, and T. Q. Quek, "Offloading in mobile edge computing: Task allocation and computational frequency scaling," *IEEE Trans. Commun.*, vol. 65, no. 8, pp. 3571-3584, Apr. 2017.

[4] C. You, K. Huang, and H. Chae, "Energy efficient mobile cloud computing powered by wireless energy transfer," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1757-1771, May

2016.

[5] S. Bi and Y.-J. A. Zhang, "Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading," *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 4177-4190, Jun. 2018.

[6] F. Wang, J. Xu, X. Wang, and S. Cui, "Joint offloading and computing optimization in wireless powered mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 1784-1797, Dec. 2017.

[7] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590-3605, Sep. 2016.

[8] S. Li, "A task offloading optimization strategy in MEC based smart cities," *Internet Technol. Lett.*, vol. 4, no. 1, p. 158, Apr. 2021.

[9] H. Guo, J. Liu, and J. Zhang, "Computation offloading for multi-access mobile edge computing in ultra-dense networks," *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 14-19, Aug. 2018.

[10] S. Sudevalayam and P. Kulkarni, "Energy harvesting sensor nodes: Survey and implications," *IEEE Commun. Surv. Tut.*, vol. 13, no. 3, pp. 443-461, Jul.-Sep. 2011.

[11] R. S. Sutton and A. G. Barto, *Reinforcement learning: An Introduction*, Cambridge, MA, USA: MIT Press, 1998.

[12] H. Zuo, G. Zhang, W. Pedrycz, V. Behbood, and J. Lu, "Fuzzy regression transfer learning in Takagi-Sugeno fuzzy models," *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 6, pp. 1795-1807, Dec. 2017.

[13] W. Liu, J. Cao, L. Yang, L. Xu, X. Qiu, and J. Li, "AppBooster: Boosting the performance of interactive mobile applications with computation offloading and parameter tuning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 6, pp. 1593-1606, Nov. 2017.

[14] I. Yoon and D. K. Noh, "Energy-Aware control of data compression and sensing rate for wireless rechargeable sensor networks," *Sensors-basel.*, vol. 18, no. 8, Aug. 2018.

[15] D. K. Noh and K. Kang, "Balanced energy allocation scheme for a solar-powered sensor system and its effects on network-wide performance," *J. Comput. Syst. Sci.*, vol. 77, no. 5, pp. 917-932, Sep. 2011.

[16] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, "Mobile-edge computing: Partial computation offloading using dynamic voltage scaling," *IEEE Trans. Commun.*, vol. 64, no. 10, pp. 4268-4282, Aug. 2016.

[17] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, "Energyoptimal mobile cloud computing under stochastic wireless channel," *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4569-4581, Aug. 2013.

[18] M. Min and L. Xiao, Y. Chen, P. Cheng, D. Wu, and W. Zhuang, "Learning-based computation offloading for IoT devices with energy harvesting," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1930-1941, Feb. 2019.

[19] L. Xiao, Y. Li, X. Huang, and X. Du, "Cloud-based malware detection game for mobile devices with offloading," *IEEE Trans. Mobile Comput.*, vol. 16, no. 10, pp. 2742-2750, Oct. 2017.

강 민 재 (Minjae Kang)

2011년 2월 : 배제대학교 컴퓨터공학과 졸업

2019년 2월 : 숭실대학교 전자공학과 박사

2019년 3월~현재 : 숭실대학교 융합특성화자유전공학부 외래교수

<관심분야> Cyber-Physical System, Mobile Network, Ubiquitous Sensor Network

[ORCID:0000-0002-1183-8752]

이 승 우 (Seungwoo Lee)

2020년 8월 : 숭실대학교 스마트시스템소프트웨어학과 졸업
2020년 9월~현재 : 숭실대학교 지능시스템학과 석사 과정
<관심분야> Cyber-Physical System, Internet of Things, Embedded System Software, AI

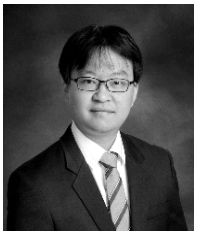
공 유 진 (Yujin Gong)

2021년 2월 : 숭실대학교 스마트시스템소프트웨어학과 졸업
2021년 3월~현재 : 숭실대학교 지능시스템학과 석사 과정
<관심분야> Cyber-Physical System, Internet of Things, Embedded System Software

김 영 현 (Younghyun Kim)

2007년 2월 : 서울대학교 컴퓨터공학부 졸업
2013년 2월 : 서울대학교 전기컴퓨터공학부 박사
2013년~2016년 : Purdue 박사 후 연구 조교수
2016년~현재 : University of Wisconsin-Madison 교수
<관심분야> Energy-Efficient Computing, Cyber-Physical Systems Security, Internet of Things
[ORCID:0000-0002-5287-9235]

윤 익 준 (Ikjune Yoon)



2006년 2월 : 전북대학교 컴퓨터공학과 졸업
2015년 2월 : 서울대학교 전기컴퓨터공학부 박사
2015년~2021년 : 숭실대학교 스마트시스템소프트웨어학과 연구교수
2021년~현재 : 경기대학교 AI컴퓨터공학부 조교수
<관심분야> Cyber-Physical System, Mobile Computing, Internet of Things
[ORCID:0000-0002-5699-162X]

노 동 건 (Dong Kun Noh)

2000년 2월 : 서울대학교 컴퓨터공학과 졸업
2002년 2월 : 서울대학교 전기컴퓨터공학부 석사
2007년 2월 : 서울대학교 전기컴퓨터공학부 박사
2007년~2010년 : Illinois at Urbana-Champaign 박사 후 연구원
2018년~2019년 : University of Wisconsin at Madison 방문연구원
2012년~현재 : 숭실대학교 지능시스템학과 교수
<관심분야> Cyber-Physical System, Mobile Computing, Internet of Things
[ORCID: 0000-0003-2068-633X]