



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2024년10월04일

(11) 등록번호 10-2712787

(24) 등록일자 2024년09월26일

- (51) 국제특허분류(Int. Cl.)
 $G06F\ 9/48$ (2018.01) $H04L\ 65/40$ (2022.01)
 $H04W\ 28/08$ (2023.01) $H04W\ 4/70$ (2018.01)
- (52) CPC특허분류
 $G06F\ 9/4856$ (2013.01)
 $G06F\ 9/4893$ (2013.01)
- (21) 출원번호 10-2021-0134362
(22) 출원일자 2021년10월08일
심사청구일자 2021년10월08일
(65) 공개번호 10-2023-0045486
(43) 공개일자 2023년04월04일
(30) 우선권주장
1020210127760 2021년09월28일 대한민국(KR)
(56) 선행기술조사문헌
한국공개특허 제10-2021-0069588호(2021.06.11.)
1부.*
미국공개특허 제2021-0004265호(2021.01.07.) 1
부.*
차량 엣지 컴퓨팅에서 로드 밸런싱을 고려한 강화
학습 기반의
마이크로레이션(<https://kiss.kstudy.com/thesis/thesis-view.asp?key=3893584>, 2021 춘계학술발표대
회 논문집 제28권 제1호(2021.05) 1부.*
*는 심사관에 의하여 인용된 문헌

- (73) 특허권자
승실대학교 산학협력단
서울특별시 동작구 상도로 369 (상도동)
- (72) 발명자
노동건
서울특별시 양천구 목동동로12길 23,1011동 207
호(목동, 목동대림아파트)
이승우
경기도 부천시 부흥로 100, 1518동 1401호(상동,
한아름마을)
강민재
서울특별시 관악구 양녕로 43.101동 201호 (봉천
동)
(74) 대리인
윤귀상

전체 청구항 수 : 총 3 항

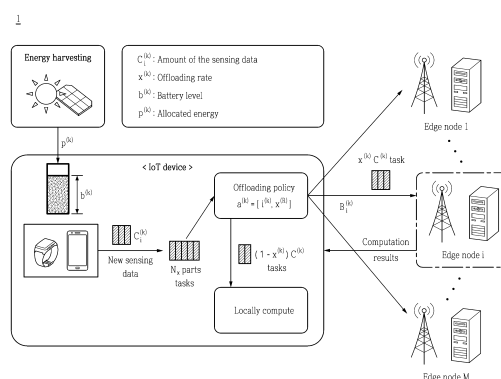
심사관 : 김종기

(54) 발명의 명칭 에너지 수집형 IoT 엣지 컴퓨팅 환경에서 강화학습 기반 동적 태스크 오프로딩 방법, 이를 수
행하기 위한 기록 매체 및 장치

(57) 요약

에너지 수집형 IoT 엣지 컴퓨팅 환경에서 강화학습 기반 동적 태스크 오프로딩 방법은, 복수의 엣지 노드와 함께
모바일 엣지 컴퓨팅(Mobile Edge Computing, MEC) 시스템을 구성하는 IoT 장치에서, 일정한 간격으로 시간을 분
할하는 단계; 분할된 시간 k에서 처리해야 할 데이터량을 센싱하는 단계; 센싱한 데이터량을 기초로 수행할 태

(뒷면에 계속)

대표도 - 도1

스크를 미리 정해진 단위로 분할하는 단계; 할당된 에너지양, 배터리 잔량 및 무선 전송속도를 기반으로 Q-러닝을 이용하여 시간 k에서 태스크를 오프로딩 할 엣지 노드 및 태스크의 오프로딩 비율을 결정하는 단계; 및 결정된 비율의 태스크를 결정된 엣지 노드에 오프로딩하고, 나머지 비율의 태스크를 로컬에서 처리하는 단계;를 포함한다. 이에 따라, 수집된 에너지를 최대한 활용해 안정적으로 오랜 기간 사용할 수 있는 높은 안정성을 제공하고, 지연시간 측면에서도 빠른 응답시간을 보여줌으로써, 높은 QoE를 제공할 수 있다.

(52) CPC특허분류

H04L 67/1012 (2022.05)

H04L 67/62 (2022.05)

H04W 28/0967 (2020.05)

H04W 4/70 (2018.02)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711126138
과제번호	2020-0-01602-002
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	정보통신방송혁신인재양성(R&D)
연구과제명	지능형 사이버 위협 대응 기술 개발 및 인력양성
기 여 율	1/1
과제수행기관명	송실대학교 산학협력단
연구기간	2021.01.01 ~ 2021.12.31

공지예외적용 : 있음

명세서

청구범위

청구항 1

복수의 엣지 노드와 함께 모바일 엣지 컴퓨팅(Mobile Edge Computing, MEC) 시스템을 구성하는 IoT 장치에서, 일정한 간격으로 시간을 분할하는 단계;

분할된 시간 k에서 처리해야 할 데이터량을 센싱하는 단계;

센싱한 데이터량을 기초로 수행할 태스크를 미리 정해진 단위로 분할하는 단계;

할당된 에너지양, 배터리 잔량 및 무선 전송속도를 기반으로 Q-러닝을 이용하여 시간 k에서 태스크를 오프로딩할 엣지 노드 및 태스크의 오프로딩 비율을 결정하는 단계; 및

결정된 비율의 태스크를 결정된 엣지 노드에 오프로딩하고, 나머지 비율의 태스크를 로컬에서 처리하는 단계;를 포함하고

상기 태스크를 오프로딩할 엣지 노드 및 태스크의 오프로딩 비율을 결정하는 단계는,

할당된 에너지양, 현재 배터리 레벨 및 각 엣지 노드의 이전 시간의 무선 전송속도를 상태로 정의하고, 오프로딩할 엣지 노드와 오프로딩 비율을 액션으로 정의하는 Q-러닝을 이용하여, 다음 시간의 오프로딩 정책을 결정하는 단계;를 더 포함하고,

상기 태스크를 오프로딩할 엣지 노드 및 태스크의 오프로딩 비율을 결정하는 단계는,

상기 결정된 오프로딩 정책에 따라 태스크 실행을 완료한 엣지 노드로부터 결과를 수신하여 평가하는 단계; 및

평가를 기반으로 Q-함수를 업데이트하는 단계;를 더 포함하고,

상기 태스크 실행을 완료한 엣지 노드로부터 결과를 수신하여 평가하는 단계는,

전체 지연시간, 에너지 활용도, 태스크 미수행 손실 및 태스크 오프로딩 이득 중 적어도 하나를 평가항목으로 활용하여 IoT 장치의 보상값을 지정하되, 상기 보상값은 다음 수학적식에 의해 산출되는, 에너지 수집형 IoT 엣지 컴퓨팅 환경에서 강화학습 기반 동적 태스크 오프로딩 방법.

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

여기서, k는 일정한 간격으로 분할된 시간, $R_i^{(k)}(x)$ 는 IoT 장치의 보상값, x 는 오프로딩 비율, $C^{(k)}$ 는 시간 k에서 처리해야할 태스크의 비트수이다. ψ 는 태스크 미수행 비용이며, $I(\varpi)$ 는 ϖ 가 참이면 1, 거짓이면 0으로 정의한다. $b^{(k+1)}$ 는 시간 k+1에서 배터리 레벨, β 는 에너지 소비의 가중치, $E^{(k)}$ 는 시간 k에서 에너지 소비, μ 는 지연시간의 가중치, $T^{(k)}$ 는 시간 k에서 지연시간이다.

청구항 2

삭제

청구항 3

삭제

청구항 4

삭제

청구항 5

삭제

청구항 6

제1항에 따른 상기 에너지 수집형 IoT 엣지 컴퓨팅 환경에서 강화학습 기반 동적 태스크 오프로딩 방법을 수행하기 위한 컴퓨터 프로그램이 기록된 컴퓨터로 판독 가능한 저장 매체.

청구항 7

복수의 엣지 노드와 함께 모바일 엣지 컴퓨팅(Mobile Edge Computing, MEC) 시스템을 구성하는 IoT 장치에서, 일정한 간격으로 시간을 분할하는 시간 분할부;

분할된 시간 k에서 처리해야 할 데이터량을 센싱하는 데이터 센싱부;

센싱한 데이터량을 기초로 수행할 태스크를 미리 정해진 단위로 분할하는 태스크 분할부;

할당된 에너지양, 배터리 잔량 및 무선 전송속도를 기반으로 Q-러닝을 이용하여 시간 k에서 태스크를 오프로딩할 엣지 노드 및 태스크의 오프로딩 비율을 결정하는 오프로딩 결정부; 및

결정된 비율의 태스크를 결정된 엣지 노드에 오프로딩하고, 나머지 비율의 태스크를 로컬에서 처리하는 로컬 처리부;를 포함하고,

상기 로컬 처리부는,

할당된 에너지양, 현재 배터리 레벨 및 각 엣지 노드의 이전 시간의 무선 전송속도를 상태로 정의하고, 오프로딩할 엣지 노드와 오프로딩 비율을 액션으로 정의하는 Q-러닝을 이용하여, 다음 시간의 오프로딩 정책을 결정하고,

상기 오프로딩 결정부는,

상기 결정된 오프로딩 정책에 따라 태스크 실행을 완료한 엣지 노드로부터 결과를 수신하여 평가하고, 평가를 기반으로 Q-함수를 업데이트하고,

상기 오프로딩 결정부는,

전체 지연시간, 에너지 활용도, 태스크 미수행 손실 및 태스크 오프로딩 이득 중 적어도 하나를 평가항목으로 활용하여 IoT 장치의 보상값을 지정하되, 상기 보상값은 다음 수학식에 의해 산출되는, 에너지 수집형 IoT 엣지 컴퓨팅 환경에서 강화학습 기반 동적 태스크 오프로딩 장치.

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

여기서, k는 일정한 간격으로 분할된 시간, $R_i^{(k)}(x)$ 는 IoT 장치의 보상값, x 는 오프로딩 비율, $C^{(k)}$ 는 시간 k에서 처리해야할 태스크의 비트수이다. ψ 는 태스크 미수행 비용이며, $I(\varpi)$ 는 ϖ 가 참이면 1, 거짓이면 0으로 정의한다. $b^{(k+1)}$ 는 시간 k+1에서 배터리 레벨, β 는 에너지 소비의 가중치, $E^{(k)}$ 는 시간 k에서 에너지 소비, μ 는 지연시간의 가중치, $T^{(k)}$ 는 시간 k에서 지연시간이다.

청구항 8

삭제

청구항 9

삭제

청구항 10

삭제

청구항 11

삭제

발명의 설명

기술 분야

- [0001] 본 발명은 에너지 수집형 IoT 엣지 컴퓨팅 환경에서 강화학습 기반 동적 태스크 오프로딩 방법, 이를 수행하기 위한 기록 매체 및 장치에 관한 것으로서, 더욱 상세하게는 최적의 오프로딩 비율과 타겟 엣지 노드를 선택하여 태양에너지 활용을 최대화하면서 사용자의 QoE(Quality Of Experience, 사용자 체감 품질)를 향상시킬 수 있는, 강화학습 기반의 동적 태스크 오프로딩 기술에 관한 것이다.

배경 기술

- [0003] 에너지 수집(EH, Energy Harvesting) 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.
- [0004] 이러한 문제점들을 해결하기 위해 EH-IoT를 이용한 많은 연구가 진행 되었다. EH은 일반적으로 주변 환경 에너지(태양, 풍력, 무선 전력 전송, 압력 등)를 에너지원으로 사용하며, 각각의 에너지원에 따라 다양한 특성들이 있다. 그 중 태양에너지는 에너지 밀도, 주기적인 수집, 높은 예측성의 장점을 가지므로, IoT 장치에서 많이 사용된다.
- [0005] 한편, 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.
- [0006] 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. 이를 IoT 응용의 QoE(quality of experience, 사용자 체감 품질)라 정의한다.
- [0007] 이러한 문제는 모바일 엣지 컴퓨팅(MEC) 기술로 해결할 수 있다. 이는 기존 IoT 장치에서 수행하던 태스크를 기지국, 액세스 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.
- [0008] 태스크 오프로딩 기법에서 해결해야 할 이슈는 다음과 같다. 연결된 후보 엣지 노드 중 타겟 엣지 노드를 선택하고, 오프로딩의 비율 결정, 즉 엣지 노드로 오프로드 할 태스크의 크기를 결정해야 하며, 무선 채널의 페이딩이나 간섭으로 인해 다양하게 변화하는 무선 링크의 전송 속도를 고려하는 것이다.
- [0009] 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.
- [0010] 그러나, 기존 연구들은 엣지 노드가 여러 개 있거나, 동적 MEC 네트워크와 같이 계속 변화하는 환경에서 실시간으로 최적의 정책을 결정하지 못한다는 한계가 있다.

선행기술문헌

특허문헌

- [0012] (특허문헌 0001) CN 111641681 A

비특허문헌

- [0013] (비특허문헌 0001) 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

발명의 내용

해결하려는 과제

- [0014] 이에, 본 발명의 기술적 과제는 이러한 점에서 착안된 것으로 본 발명의 목적은 에너지 수집형 IoT 엣지 컴퓨팅 환경에서 강화학습 기반 동적 태스크 오프로딩 방법을 제공하는 것이다.
- [0015] 본 발명의 다른 목적은 상기 에너지 수집형 IoT 엣지 컴퓨팅 환경에서 강화학습 기반 동적 태스크 오프로딩 방법을 수행하기 위한 컴퓨터 프로그램이 기록된 기록 매체를 제공하는 것이다.
- [0016] 본 발명의 또 다른 목적은 상기 에너지 수집형 IoT 엣지 컴퓨팅 환경에서 강화학습 기반 동적 태스크 오프로딩 방법을 수행하기 위한 장치를 제공하는 것이다.

과제의 해결 수단

- [0018] 상기한 본 발명의 목적을 실현하기 위한 일 실시예에 따른 에너지 수집형 IoT 엣지 컴퓨팅 환경에서 강화학습 기반 동적 태스크 오프로딩 방법은, 복수의 엣지 노드와 함께 모바일 엣지 컴퓨팅(Mobile Edge Computing, MEC) 시스템을 구성하는 IoT 장치에서, 일정한 간격으로 시간을 분할하는 단계; 분할된 시간 k 에서 처리해야 할 데이터량을 센싱하는 단계; 센싱한 데이터량을 기초로 수행할 태스크를 미리 정해진 단위로 분할하는 단계; 할당된 에너지양, 배터리 잔량 및 무선 전송속도를 기반으로 Q-러닝을 이용하여 시간 k 에서 태스크를 오프로딩 할 엣지 노드 및 태스크의 오프로딩 비율을 결정하는 단계; 및 결정된 비율의 태스크를 결정된 엣지 노드에 오프로딩하고, 나머지 비율의 태스크를 로컬에서 처리하는 단계;를 포함한다.
- [0019] 본 발명의 실시예에서, 상기 태스크를 오프로딩 할 엣지 노드 및 태스크의 오프로딩 비율을 결정하는 단계는, 태스크 실행을 완료한 엣지 노드로부터 결과를 수신하여 평가하는 단계; 및 평가를 기반으로 Q-함수를 업데이트 하는 단계;를 더 포함할 수 있다.
- [0020] 본 발명의 실시예에서, 상기 태스크 실행을 완료한 엣지 노드로부터 결과를 수신하여 평가하는 단계는, 전체 지연시간, 에너지 활용도, 태스크 미수행 손실 및 태스크 오프로딩 이득 중 적어도 하나를 평가항목으로 활용할 수 있다.
- [0021] 본 발명의 실시예에서, 상기 태스크를 오프로딩 할 엣지 노드 및 태스크의 오프로딩 비율을 결정하는 단계는, 전체 지연시간, 에너지 활용도, 태스크 미수행 손실 및 태스크 오프로딩 이득 중 적어도 하나를 기초로 IoT 장치의 보상값을 결정하는 단계;를 더 포함할 수 있다.
- [0022] 본 발명의 실시예에서, 상기 태스크를 오프로딩 할 엣지 노드 및 태스크의 오프로딩 비율을 결정하는 단계는, 할당된 에너지양, 현재 배터리 레벨 및 각 엣지 노드의 이전 시간의 무선 전송속도를 상태로 정의하고, 오프로딩할 엣지 노드와 오프로딩 비율을 액션으로 정의하는 Q-러닝을 이용하여, 다음 시간의 오프로딩 정책을 결정하는 단계;를 더 포함할 수 있다.
- [0023] 상기한 본 발명의 다른 목적을 실현하기 위한 일 실시예에 따른 컴퓨터로 판독 가능한 저장 매체에는, 상기 에너지 수집형 IoT 엣지 컴퓨팅 환경에서 강화학습 기반 동적 태스크 오프로딩 방법을 수행하기 위한 컴퓨터 프로그램이 기록되어 있다.
- [0024] 상기한 본 발명의 또 다른 목적을 실현하기 위한 일 실시예에 따른 에너지 수집형 IoT 엣지 컴퓨팅 환경에서 강화학습 기반 동적 태스크 오프로딩 장치는, 복수의 엣지 노드와 함께 모바일 엣지 컴퓨팅(Mobile Edge Computing, MEC) 시스템을 구성하는 IoT 장치에서, 일정한 간격으로 시간을 분할하는 시간 분할부; 분할된 시간 k 에서 처리해야 할 데이터량을 센싱하는 데이터 센싱부; 센싱한 데이터량을 기초로 수행할 태스크를 미리 정해진 단위로 분할하는 태스크 분할부; 할당된 에너지양, 배터리 잔량 및 무선 전송속도를 기반으로 Q-러닝을 이용하여 시간 k 에서 태스크를 오프로딩 할 엣지 노드 및 태스크의 오프로딩 비율을 결정하는 오프로딩 결정부; 및 결정된 비율의 태스크를 결정된 엣지 노드에 오프로딩하고, 나머지 비율의 태스크를 로컬에서 처리하는

로컬 처리부;를 포함한다.

- [0025] 본 발명의 실시예에서, 상기 오프로딩 결정부는, 태스크 실행을 완료한 엣지 노드로부터 결과를 수신하여 평가하고, 평가를 기반으로 Q-함수를 업데이트할 수 있다.
- [0026] 본 발명의 실시예에서, 상기 오프로딩 결정부는, 전체 지연시간, 에너지 활용도, 태스크 미수행 손실 및 태스크 오프로딩 이득 중 적어도 하나를 평가항목으로 활용할 수 있다.
- [0027] 본 발명의 실시예에서, 상기 오프로딩 결정부는, 전체 지연시간, 에너지 활용도, 태스크 미수행 손실 및 태스크 오프로딩 이득 중 적어도 하나를 기초로 IoT 장치의 보상값을 결정할 수 있다.
- [0028] 본 발명의 실시예에서, 상기 오프로딩 결정부는, 할당된 에너지양, 현재 배터리 레벨 및 각 엣지 노드의 이전 시간의 무선 전송속도를 상태로 정의하고, 오프로딩할 엣지 노드와 오프로딩 비율을 액션으로 정의하는 Q-러닝을 이용하여, 다음 시간의 오프로딩 정책을 결정할 수 있다.

발명의 효과

- [0030] 이와 같은 에너지 수집형 IoT 엣지 컴퓨팅 환경에서 강화학습 기반 동적 태스크 오프로딩 방법에 따르면, IoT 장치는 현재 배터리 레벨, 무선 전송속도와 예측된 에너지 수집 모델에 따라 오프로딩할 타겟 엣지 노드와 오프로딩 비율을 선택한다.
- [0031] 또한, 전이 학습(Transfer Learning, TL)을 사용하여 초기 단계의 학습 속도를 가속화하여 오프로딩 성능을 향상시키고, 에너지 수집형 엣지 컴퓨팅 환경에서 사용자의 QoE(응답지연시간, 에너지 활용도)를 향상시킬 수 있다.

도면의 간단한 설명

- [0033] 도 1은 본 발명의 IoT 장치와 M 개의 엣지 노드로 구성된 모바일 엣지 컴퓨팅(Mobile Edge Computing, MEC) 시스템의 개요를 보여주는 개념도이다.
- 도 2는 본 발명의 일 실시예에 따른 에너지 수집형 IoT 엣지 컴퓨팅 환경에서 강화학습 기반 동적 태스크 오프로딩 장치의 블록도이다.
- 도 3은 수집에너지를 각 단위 시간 슬롯에 할당하는 것을 설명하기 위한 도면이다.
- 도 4는 본 발명에서 사용하는 사용하는 Q-learning 기반 태스크 오프로딩 기법(QTO)을 보여주는 알고리즘이다.
- 도 5는 본 발명의 성능을 검증하기 위해 IoT 개수를 다르게 하여 기존 알고리즘들과 정전시간을 비교한 시뮬레이션 결과이다.
- 도 6은 본 발명의 성능을 검증하기 위해 IoT 개수를 다르게 하여 기존 알고리즘들과 지연시간을 비교한 시뮬레이션 결과이다.
- 도 7은 본 발명의 성능을 검증하기 위해 IoT 개수를 다르게 하여 기존 알고리즘들과 태스크 미수행 측면을 비교한 시뮬레이션 결과이다.
- 도 8은 본 발명의 성능을 검증하기 위해 태스크의 크기를 다르게 하여 기존 알고리즘들과 정전시간을 비교한 시뮬레이션 결과이다.
- 도 9는 본 발명의 성능을 검증하기 위해 태스크의 크기를 다르게 하여 기존 알고리즘들과 지연시간을 비교한 시뮬레이션 결과이다.
- 도 10은 본 발명의 성능을 검증하기 위해 태스크의 크기를 다르게 하여 기존 알고리즘들과 태스크 미수행 측면을 비교한 시뮬레이션 결과이다.
- 도 11은 본 발명의 일 실시예에 따른 에너지 수집형 IoT 엣지 컴퓨팅 환경에서 강화학습 기반 동적 태스크 오프로딩 방법의 흐름도이다.

발명을 실시하기 위한 구체적인 내용

- [0034] 후술하는 본 발명에 대한 상세한 설명은, 본 발명이 실시될 수 있는 특정 실시예를 예시로서 도시하는 첨부 도면을 참조한다. 이들 실시예는 당업자가 본 발명을 실시할 수 있기에 충분하도록 상세히 설명된다. 본 발명의

다양한 실시예는 서로 다르지만 상호 배타적일 필요는 없음이 이해되어야 한다. 예를 들어, 여기에 기재되어 있는 특정 형상, 구조 및 특성은 일 실시예에 관련하여 본 발명의 정신 및 범위를 벗어나지 않으면서 다른 실시예로 구현될 수 있다. 또한, 각각의 개시된 실시예 내의 개별 구성요소의 위치 또는 배치는 본 발명의 정신 및 범위를 벗어나지 않으면서 변경될 수 있음이 이해되어야 한다. 따라서, 후술하는 상세한 설명은 한정적인 의미로서 취하려는 것이 아니며, 본 발명의 범위는, 적절하게 설명된다면, 그 청구항들이 주장하는 것과 균등한 모든 범위와 더불어 첨부된 청구항에 의해서만 한정된다. 도면에서 유사한 참조부호는 여러 측면에 걸쳐서 동일하거나 유사한 기능을 지칭한다.

- [0035] 이하, 도면들을 참조하여 본 발명의 바람직한 실시예들을 보다 상세하게 설명하기로 한다.
- [0036] 도 1은 IoT 장치와 M 개의 엣지 노드로 구성된 모바일 엣지 컴퓨팅(Mobile Edge Computing, MEC) 시스템의 개요를 보여주는 개념도이다.
- [0037] 도 1에서, 모바일 엣지 컴퓨팅(Mobile Edge Computing, MEC) 시스템(1, 이하 MEC 시스템)을 구성하는 IoT 장치는 예를 들어, 스마트 시계와 스마트폰 등의 장치를 의미하며, 태양광 패널과 같은 에너지 수집 모듈과 배터리가 장착되어 있다.
- [0038] 이러한 IoT 장치는 보통 센싱된 데이터를 로컬에서 처리한다. 하지만, 처리해야 되는 태스크가 얼굴 인식 응용과 같은 고도화된 작업이라면 로컬에서 처리하는데 지연시간과 에너지 측면에서 제한적이다.
- [0039] 따라서, 얼굴 인식 응용과 같은 연산 집약적 태스크는 엣지 노드로 처리해야 할 태스크를 전송하고 처리된 결과를 받음으로써, IoT 장치의 에너지 절약하고 및 지연시간에 도움이 될 수 있다. 본 발명에서는 이렇게 엣지 노드로 태스크를 전달하여 처리하는 것을 태스크 오프로딩이라 정의한다.
- [0040] 도 2는 본 발명의 일 실시예에 따른 에너지 수집형 IoT 엣지 컴퓨팅 환경에서 강화학습 기반 동적 태스크 오프로딩 장치의 블록도이다.
- [0041] 본 발명에 따른 에너지 수집형 IoT 엣지 컴퓨팅 환경에서 강화학습 기반 동적 태스크 오프로딩 장치(10, 이하 장치)는 에너지 수집형 IoT 엣지 컴퓨팅 환경에서, 수집에너지 활용을 최대화 하면서 사용자의 QoE를 향상시킬 수 있는, 강화학습 기반의 동적 태스크 오프로딩 기법을 제안한다.
- [0042] 본 발명에서 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.
- [0043] 결과적으로, 본 발명은 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.
- [0044] 도 2를 참조하면, 본 발명에 따른 장치(10)는 시간 분할부(110), 데이터 센싱부(130), 태스크 분할부(150), 오프로딩 결정부(170) 및 로컬 처리부(190)를 포함한다.
- [0045] 본 발명의 상기 장치(10)는 MEC 시스템(1)을 구성하는 IoT 장치이거나, IoT 장치의 일부를 구성할 수 있다.
- [0046] 본 발명의 상기 장치(10)는 에너지 수집형 IoT 엣지 컴퓨팅 환경에서 강화학습 기반 동적 태스크 오프로딩 방법을 수행하기 위한 소프트웨어(애플리케이션)가 설치되어 실행될 수 있으며, 상기 시간 분할부(110), 상기 데이터 센싱부(130), 상기 태스크 분할부(150), 상기 오프로딩 결정부(170) 및 상기 로컬 처리부(190)의 구성은 상기 장치(10)에서 실행되는 상기 에너지 수집형 IoT 엣지 컴퓨팅 환경에서 강화학습 기반 동적 태스크 오프로딩 방법을 수행하기 위한 소프트웨어에 의해 제어될 수 있다.
- [0047] 또한, 상기 시간 분할부(110), 상기 데이터 센싱부(130), 상기 태스크 분할부(150), 상기 오프로딩 결정부(170) 및 상기 로컬 처리부(190)의 구성은 통합 모듈로 형성되거나, 하나 이상의 모듈로 이루어 질 수 있다. 그러나, 이와 반대로 각 구성은 별도의 모듈로 이루어질 수도 있다.
- [0048] 상기 장치(10)는 이동성을 갖거나 고정될 수 있다. 상기 장치(10)는, 디바이스(device), 기구(apparatus), 단말(terminal), UE(user equipment), MS(mobile station), 무선기기(wireless device), 휴대기기(handheld device) 등 다른 용어로 불릴 수 있다.
- [0049] 상기 장치(10)는 운영체제(Operation System; OS), 즉 시스템을 기반으로 다양한 소프트웨어를 실행하거나 제작

할 수 있다. 상기 운영체제는 소프트웨어가 장치의 하드웨어를 사용할 수 있도록 하기 위한 시스템 프로그램으로서, 안드로이드 OS, iOS, 윈도우 모바일 OS, 바다 OS, 심비안 OS, 블랙베리 OS 등 모바일 컴퓨터 운영체제 및 윈도우 계열, 리눅스 계열, 유닉스 계열, MAC, AIX, HP-UX 등 컴퓨터 운영체제를 모두 포함할 수 있다.

도 1 및 도 2를 참조하면, 먼저 상기 장치(10)의 시간 분할부(110)가 일정한 간격으로 시간을 나누고 이는 $k(k \in 1, 2, \dots)$ 로 정한다. 상기 데이터 센싱부(130)는 시간 k 에서 C^k 비트만큼 처리해야 할 데이터를 센싱한다.

상기 태스크 분할부(150)는 예를 들어, Computation Partition Scheme을 이용해 태스크를 N_x 단위로 나눈다.

상기 오프로딩 결정부(170)는 무선 전송속도 $B_i^{(k)} (1 \leq i \leq M)$ 를 기반으로 시간 k 에서 태스크를 오프로딩 할 엣지 노드 i 를 선택하고, 오프로딩 비율 $x^{(k)} (x^{(k)} \in \{l/N_x\}_{0 \leq l \leq N_x})$ 를 결정한다.

구체적으로, IoT 장치는 $x^{(k)} = 0$ 일 때, 로컬에서 모든 태스크를 처리하고, $x^{(k)} = 1$ 일 때 모든 태스크를 오프로딩한다. 따라서, IoT 장치는 $x^{(k)} C^{(k)}$ 비트를 타겟 엣지 노드 i 로 오프로딩 하고, 나머지 $(1 - x^{(k)}) C^{(k)}$ 비트는 로컬에서 처리한다 $(0 < x^{(k)} < 1)$.

상기 오프로딩 결정부(170)는 시간 k 에서 오프로딩 정책 $a^{(k)} = [i^{(k)}, x^{(k)}] (a \in A)$ 을 선택한다. 상기 로컬 처리부(190)는 결정된 비율의 태스크를 결정된 엣지 노드에 오프로딩하고, 나머지 비율의 태스크를 로컬에서 처리한다.

각각의 기호는 아래의 표 1에서 정의한다. 또한, 시간 표기가 의미가 없는 경우 시간 인덱스 k 는 생략한다.

[표 1]

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

본 발명에서는, 태스크를 로컬에서 수행 시 소모되는 에너지와 지연시간을 엣지 노드로의 오프로딩 시 소모되는 양 등과 비교함으로써, 각 태스크의 오프로딩 여부 및 타겟 엣지 노드를 선택한다.

결과적으로, IoT 장치의 태스크 처리 지연시간과 정전시간을 감소시켜 사용자의 QoE를 증가시킬 수 있게 된다.

[0061] 이하에서는, 태양에너지 수집 및 할당 모델을 설명한다. 구체적으로, 노드가 각 시간 슬롯에 에너지를 얼마나 할당할지에 관한 모델링이며, 이는 에너지가 수집되지 않는 밤 시간대에도 안정적인 동작 수행을 가능케 하도록 설계되었다.

[0062] 이렇게 각 노드의 타임슬롯 별로 할당된 에너지양을 기반으로, 각각 로컬에서 태스크를 처리할 때 지연시간과 에너지 소모량/잔여 에너지양 예측 방법, 태스크 오프로딩시 지연시간과 에너지 소모량/잔여 에너지양 예측 방법에 대해 설명한다.

[0063] IoT 장치는 주변 환경(태양, 바람, 무선 전력 전송, 압력)에서 에너지를 수집할 수 있으며, 그 중 태양에너지는 높은 에너지 밀도를 가지고 있고, 주기적으로 수집되며, 정확한 예측이 가능하여 많은 분야에서 사용되고 있다.

[0064] 따라서, 본 발명에서는 에너지 수집 자원으로 태양에너지를 사용한다. 그러나, 태양에너지는 일몰 이후 수집되는 에너지가 거의 없어 노드의 정전을 유발할 수 있으며, 낮에는 에너지 수집량이 많아 저장 가능한 배터리 용량을 넘어서는 잉여 에너지가 발생할 수 있다.

[0065] 따라서, 시간과 무관하게 균일한 에너지 소비를 위해 균형적으로 에너지를 할당하는 방식이 필요하다. 이러한 점들을 고려한 균형적인 에너지 할당 방식을 제안하며, 도 3과 같이 수집에너지를 각 단위 시간 슬롯에 효율적이고 균등하게 할당할 수 있다.

[0066] 아래의 수학적 식 1은 에너지 할당 기법에 사용하는 수학적식이다.

[0067] [수학적 식 1]

[0068]

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

[0070] 여기서, $p^{(k)}$ 는 시간 슬롯 k에서 태스크 처리를 위해 할당된 에너지양이며, $E_{\text{alloc}}^{(k)}$ 는 시간 슬롯 k에 할당된 총 수집에너지이며, $E_{\text{sys}}^{(k)}$ 는 태스크 처리에 사용되는 에너지를 제외한 모든 소비 에너지이다.

[0071] 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 처리 하는데 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.

[0072] 또한, 에너지 소비를 제어하기 위해 동적 전압 및 주파수 스케일링 기술을 사용할 수 있는데, IoT 장치는 사이클 m ($m \in \{1, 2, \dots, (1-x^{(k)})C^{(k)}N\}$) 에서 주파수 f_m 을 조절해 에너지 소비를 제어한다.

[0073] 시간 슬롯 k에서 로컬 컴퓨팅 실행 대기 시간은 $T_0^{(k)}$ 로 표기하고, 아래의 수학적 식 2와 같이 정의한다.

[0074] [수학적 식 2]

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

[0075]

[0077] IoT 장치는 시간 k 에서 $E_0^{(k)}$ 에너지를 소비하며 아래의 수학적식 3과 같다.

[0078] [수학적식 3]

[0079]

$$E_0^{(k)} = \sum_{m=1}^{(1-x^{(k)})C^{(k)}N} \zeta f_m^2$$

[0081] 여기서, ζ 는 칩 아키텍처에 따른 유효한 커패시턴스 계수이다. 이는 동적 주파수 및 전압 스케일링을 가진 IoT 장치에 적용될 수 있다.

[0082] IoT 장치는 태스크를 오프로딩 할 때 무선 전송속도 $B_i^{(k)}$ 에 따라 전송 시간이 달라지며, 엣지 노드 i 의 CPU에 따라 태스크의 처리 속도가 달라진다. 시간 슬롯 k 에서 엣지 노드 i 로 태스크를 오프로딩 할 때, $x^{(k)}C^{(k)}$ 비트를 처리하는 시간은 $T_i^{(k)}$ 로 표기하고, 아래의 수학적식 4와 같다.

[0083] [수학적식 4]

[0084]

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

[0086] IoT 장치는 태스크를 오프로딩할 때 에너지 $E_i^{(k)}$ 를 소비한다. 이는 전송 전력 P 와 전송 시간에 따라 다르며, 아래의 수학적식 5와 같다.

[0087] [수학적식 5]

[0088]

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

[0090] 시간 k 에서 IoT 장치가 오프로딩을 포함하여 태스크를 위해 소비하는 총 에너지는 $E^{(k)} = E_0^{(k)} + E_i^{(k)}$ 로 정의한다. 시간 k 에서 배터리 레벨은 $b^{(k)}$ 로 정의하고, 아래의 수학적식 6과 같이 계산한다.

[0091] [수학적식 6]

[0092]

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

[0094] 상기 모델을 기반으로 본 발명에서는 태스크의 오프로딩 비율과 타겟 엣지 노드를 선택한다. 본 발명은 기존의 최적화 방법을 이용한 기법이 아닌 동적인 환경에 대응하기 적합한 Q-learning을 이용해 최적의 오프로딩 비율과 타겟 엣지 노드를 선택한다.

[0095] 도 4를 참조하면, 상기 오프로딩 결정부(170)에서 사용하는 Q-learning 기반 태스크 오프로딩 기법(QTO)을 설명하며, 요약하면 다음과 같다.

- [0096] 1) 태스크 오프로딩을 위한 MDP(Markov Decision Processes)에 관해 설명하고, 2) 매개 변수의 초기값 설정에 관해 설명하며, 3) 상태와 액션을 정의하고, 4) 실행된 결과를 받아 평가하고, 5) Q-함수를 업데이트한다.
- [0097] 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.
- [0098] 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.
- [0099] 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 TL을 사용한다.
- [0100] 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.
- [0101] Q-learning을 하기 위해 상태-액션 쌍이 필요하며, 상태는 할당된 에너지양, 배터리 레벨, 무선 전송속도로 정의된다. 구체적으로, IoT 장치는 수학식 5에 따라 할당된 에너지양 $p^{(k)}$ 를 예측하고, 현재 배터리 레벨 $b^{(k)}$ 와 엣지 노드 M의 이전 무선 전송속도 $B_1^{(k-1)}, \dots, B_M^{(k-1)}$ 를 상태로 정의하고, $s^{(k)} = [B_1^{(k-1)}, \dots, B_M^{(k-1)}, p^{(k)}, b^{(k)}]$ 로 표현할 수 있다.
- [0102] 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.
- [0103] 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.
- [0104] 본 발명에서는 decaying ϵ -greedy 정책을 적용해 초반에는 탐색(exploration)의 비율을 크게 하고, Q-learning이 반복될 수록 이용(exploitation)의 비율을 늘리는 방법을 사용한다.
- [0105] 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 엣지 노드 i로 오프로딩하고, $(1 - x^{(k)}) C^{(k)}$ 비트는 로컬에서 컴퓨팅한다.
- [0106] 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.
- [0107] 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 $T^{(k)} = \max\{T_0^{(k)}, T_i^{(k)}\}$ 로 정의하며, IoT 장치의 에너지가 부족한 경우에는 태스크를 수행하지 못하기 때문에 태스크 미수행에 대한 평가를 해야 한다.
- [0108] 태스크 미수행 비용은 ψ 로 정의하며, $I(\omega)$ 는 ω 가 참이면 1, 거짓이면 0으로 정의한다. β 는 에너지 소비의

가중치, μ 는 지연시간의 가중치이다. IoT 장치의 보상은 $R_i^{(k)}(x)$ 로 표시되며, 이 값은 전체 지연시간, 에너지 활용도, 태스크 미수행 손실, 태스크 오프로딩 이득에 달라지며 아래의 수학적식 7과 같다.

[수학적식 7]

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

평가가 완료된 후 평가를 기반으로 Q-함수를 업데이트해서 정책의 질을 높일 수 있다. 이를 반복할수록 최적의 정책을 찾을 수 있다. 구체적으로, IoT 장치가 상태 $s^{(k)}$ 에서 $x^{(k)}C^{(k)}$ 비트를 엣지 노드 i로 오프로딩하면 상태는 $s^{(k+1)}$ 로 전환된다. 이를 기반으로, 지속적인 오프로딩 경험

$(s^{(k)}, a^{(k)}, R^{(k)}, s^{(k+1)})$ 에 의해 IoT 장치는 Q-함수를 업데이트하며 아래의 수학적식 8과 같이 표현할 수 있다.

[수학적식 8]

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

여기서, 학습 비율 α 는 현재 오프로딩의 가중치이며, γ 는 미래 보상의 감소 계수이다. QT0 기법의 알고리즘은 도 4에 요약되어 있다. 이 기법은 충분한 시간에 걸쳐 시행착오를 통해 최적의 태스크 오프로딩 정책을 달성할 수 있다.

이하에서는, 본 발명의 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 여러 번 반복하여 평균을 구하였다. 데이터의 생성 속도는 1.0mb/s, 1.5mb/s, 2.0mb/s이고, 에너지 수집 모델은 에너지 할당 기법을 사용하였다. 또한, 제안기법의 파라미터인 $\psi, \beta, \mu, \alpha, \gamma, \epsilon$ 은 각각 10, 0.7, 1, 0.9, 0.5, 0.1로 설정하였다.

시뮬레이션은 제안기법인 QT0와 에너지 수집에 대한 고려가 없는 일반적인 Q-learning 기반 기법인 Q와 non-offloading scheme, 휴리스틱 알고리즘을 사용해 태스크를 오프로딩하는 DACO, HGOS를 포함하여 총 5 가지를 비교군으로 IoT 장치의 정전시간, 지연시간, 태스크 미수행 측면에서 진행하였다.

도 4 내지 도 7에서 볼 수 있듯이, 10개의 IoT 장치의 경우 Non-offloading scheme과 비교해 Q의 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, Q와 비교해 본 발명의 제안 기법인 QT0는 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. 이는 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.

- [0122] 도 8 내지 도 10은 처리할 태스크의 크기를 다르게 하여 실험한 결과이다. 제안기법이 태스크의 크기에 따라서도 정전시간과 지연시간, 태스크 미수행 측면에서 우수한 것을 볼 수 있으며, 특별히 HGOS는 2Mbyte의 태스크 크기에서 적절한 오프로딩을 선택을 하지 못해 태스크 미수행 측면에서 좋지 않은 결과를 보인다.
- [0124] 도 11은 본 발명의 일 실시예에 따른 에너지 수집형 IoT 엣지 컴퓨팅 환경에서 강화학습 기반 동적 태스크 오프로딩 방법의 흐름도이다.
- [0125] 본 실시예에 따른 에너지 수집형 IoT 엣지 컴퓨팅 환경에서 강화학습 기반 동적 태스크 오프로딩 방법은, 도 1의 MEC 시스템(1) 및 도 2의 장치(10)와 실질적으로 동일한 구성에서 진행될 수 있다.
- [0126] 따라서, 도 1의 MEC 시스템(1) 및 도 2의 장치(10)와 동일한 구성요소는 동일한 도면부호를 부여하고, 반복되는 설명은 생략한다. 또한, 본 실시예에 따른 에너지 수집형 IoT 엣지 컴퓨팅 환경에서 강화학습 기반 동적 태스크 오프로딩 방법은 에너지 수집형 IoT 엣지 컴퓨팅 환경에서 강화학습 기반 동적 태스크 오프로딩을 수행하기 위한 소프트웨어(애플리케이션)에 의해 실행될 수 있다.
- [0127] 도 11을 참조하면, 본 실시예에 따른 에너지 수집형 IoT 엣지 컴퓨팅 환경에서 강화학습 기반 동적 태스크 오프로딩 방법은, 복수의 엣지 노드와 함께 MEC 시스템을 구성하는 IoT 장치에서, 일정한 간격으로 시간을 분할한다(단계 S10).
- [0128] 분할된 시간 k 에서 처리해야 할 데이터량을 센싱한다(단계 S20). 센싱한 데이터량을 기초로 수행할 태스크를 미리 정해진 단위로 분할한다(단계 S30). 데이터량은 비트 단위로 센싱할 수 있으며, 예를 들어, 태스크는 Computation Partition Scheme을 이용해 N_x 단위로 나눌 수 있다.
- [0129] 이후, 할당된 에너지량, 배터리 잔량 및 무선 전송속도를 기반으로 Q-러닝을 이용하여 시간 k 에서 태스크를 오프로딩 할 엣지 노드 및 태스크의 오프로딩 비율을 결정한다(단계 S40).
- [0130] 예를 들어, 오프로딩 비율이 0일 때, 로컬에서 모든 태스크를 처리하고, 오프로딩 비율이 1일 때는 모든 태스크를 오프로딩 한다.
- [0131] 일 실시예에서, 태스크 실행을 완료한 엣지 노드로부터 결과를 수신하여 평가하고, 평가를 기반으로 Q-함수를 업데이트한다. 이때, 전체 지연시간, 에너지 활용도, 태스크 미수행 손실 및 태스크 오프로딩 이득 중 적어도 하나를 평가항목으로 활용할 수 있다.
- [0132] 또한, 전체 지연시간, 에너지 활용도, 태스크 미수행 손실 및 태스크 오프로딩 이득 중 적어도 하나를 기초로 IoT 장치의 보상값을 결정할 수 있다.
- [0133] 결정된 비율의 태스크를 결정된 엣지 노드에 오프로딩하고, 나머지 비율의 태스크를 로컬에서 처리한다(단계 S50).
- [0134] 본 발명은 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.
- [0135] 결과적으로, 본 발명은 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.
- [0136] 이와 같은, 에너지 수집형 IoT 엣지 컴퓨팅 환경에서 강화학습 기반 동적 태스크 오프로딩 방법은 애플리케이션으로 구현되거나 다양한 컴퓨터 구성요소를 통하여 수행될 수 있는 프로그램 명령어의 형태로 구현되어 컴퓨터 판독 가능한 기록 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능한 기록 매체는 프로그램 명령어, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다.
- [0137] 상기 컴퓨터 판독 가능한 기록 매체에 기록되는 프로그램 명령어는 본 발명을 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 분야의 당업자에게 공지되어 사용 가능한 것일 수도 있다.
- [0138] 컴퓨터 판독 가능한 기록 매체의 예에는, 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체, CD-ROM, DVD와 같은 광기록 매체, 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 ROM, RAM, 플래시 메모리 등과 같은 프로그램 명령어를 저장하고 수행하도록 특별히 구성된 하드웨어

어 장치가 포함된다.

[0139] 프로그램 명령어의 예에는, 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드도 포함된다. 상기 하드웨어 장치는 본 발명에 따른 처리를 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

[0140] 이상에서는 실시예들을 참조하여 설명하였지만, 해당 기술 분야의 숙련된 당업자는 하기의 특허 청구의 범위에 기재된 본 발명의 사상 및 영역으로부터 벗어나지 않는 범위 내에서 본 발명을 다양하게 수정 및 변경시킬 수 있음을 이해할 수 있을 것이다.

산업상 이용가능성

[0142] 본 발명은 태양 에너지 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. 기존에 많이 사용되는 Lyapunov 최적화 또는 convex 최적화 기법은 등은 비용(즉, 에너지 사용량 또는 작업 지연 시간) 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. 본 발명은 이러한 한계점이 없는 강화학습(현재 시점의 해당 노드의 정보만 필요함)을 이용해 태스크 오프로딩 기법을 제안하였으며, 이는 작업이 반복될수록 최적의 동작을 하게 된다.

[0143] 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.

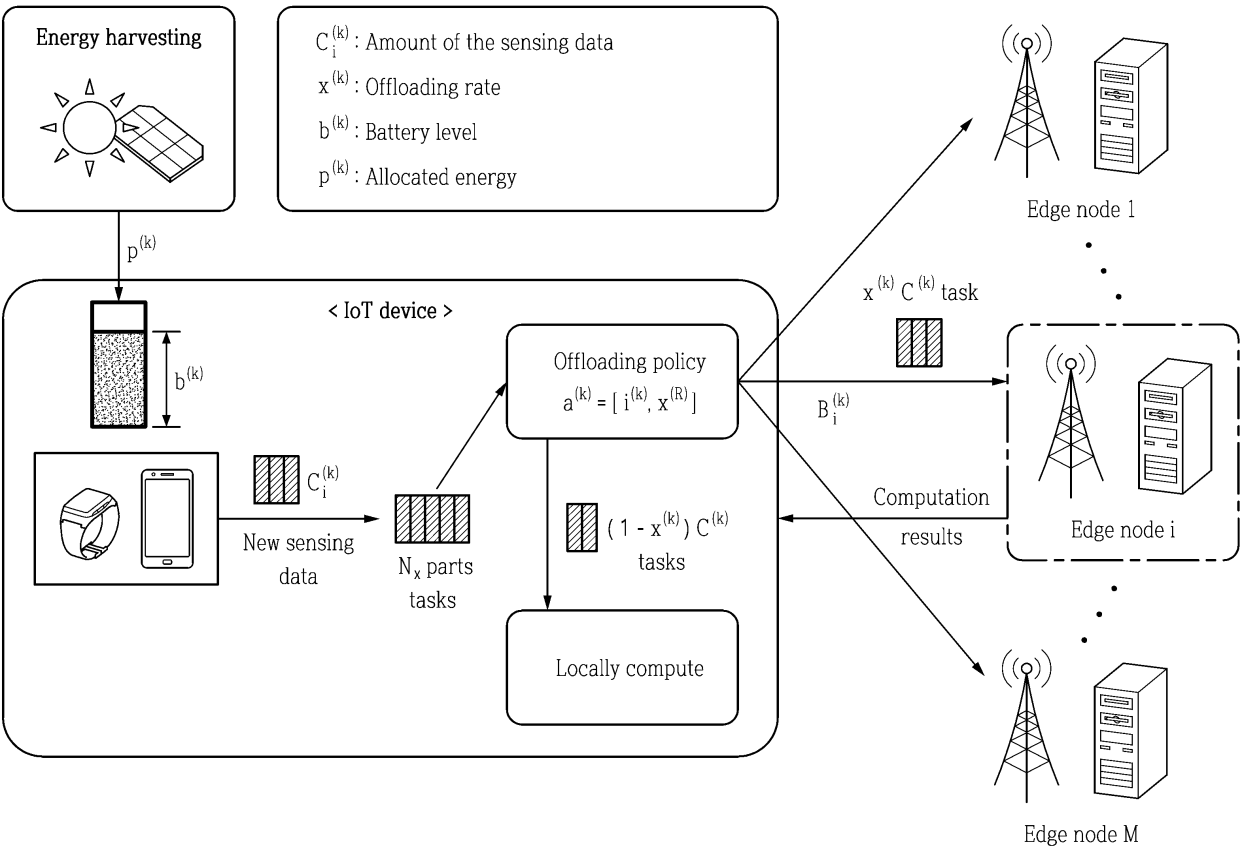
부호의 설명

- [0145] 1: 모바일 엣지 컴퓨팅(Mobile Edge Computing, MEC) 시스템
10: 강화학습 기반 동적 태스크 오프로딩 장치
110: 시간 분할부
130: 데이터 센싱부
150: 태스크 분할부
170: 오프로딩 결정부
190: 로컬 처리부

도면

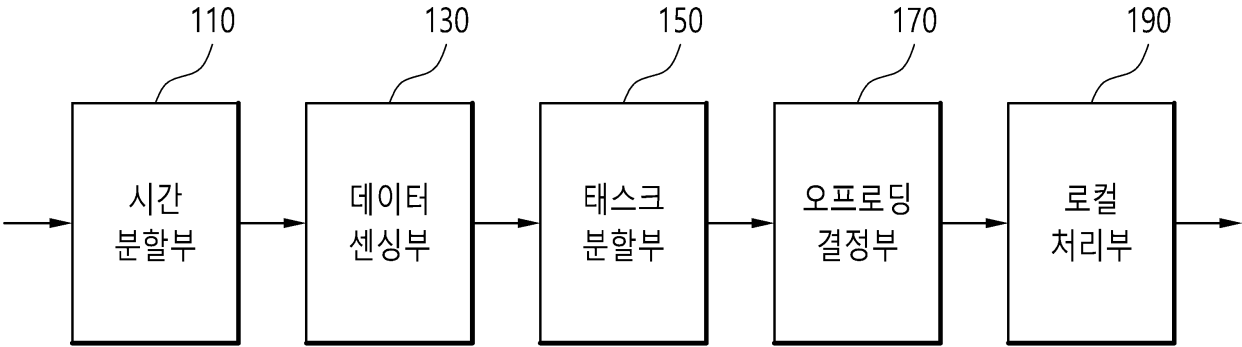
도면1

1

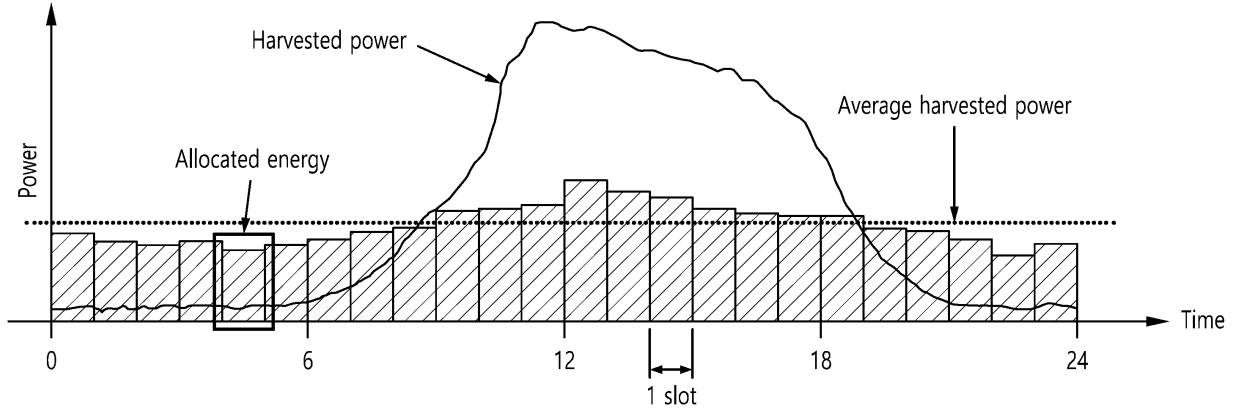


도면2

10



도면3

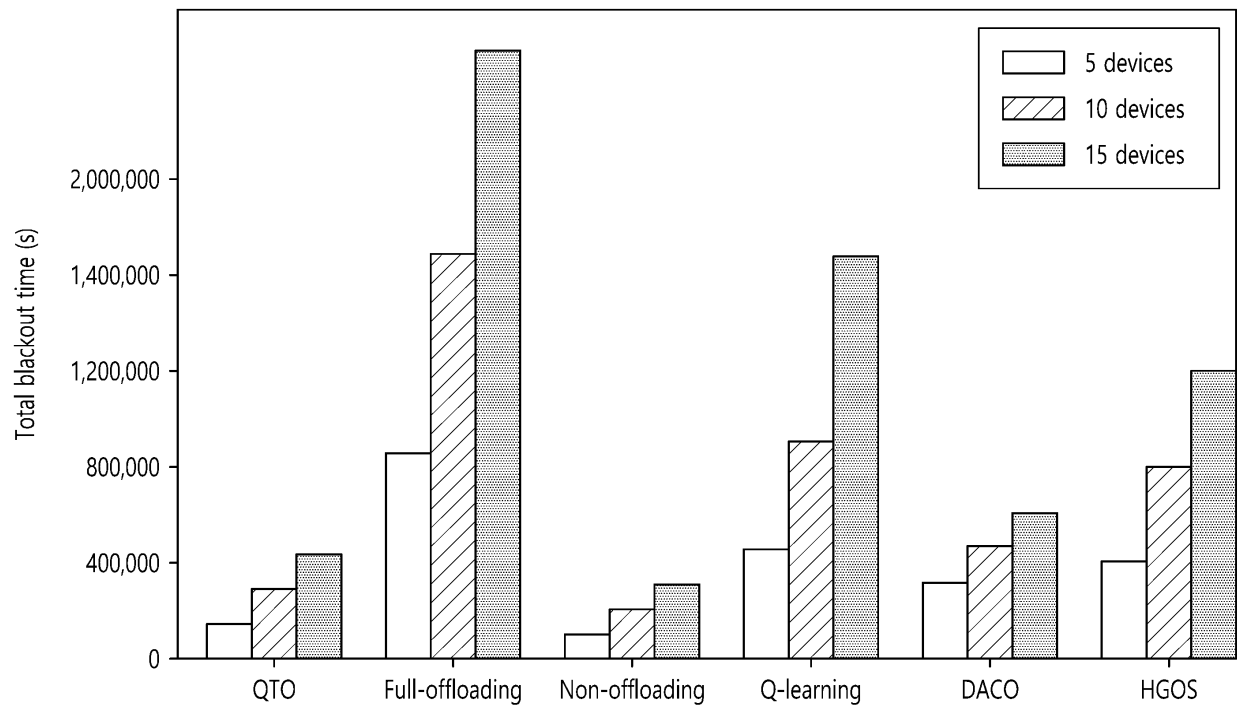


도면4

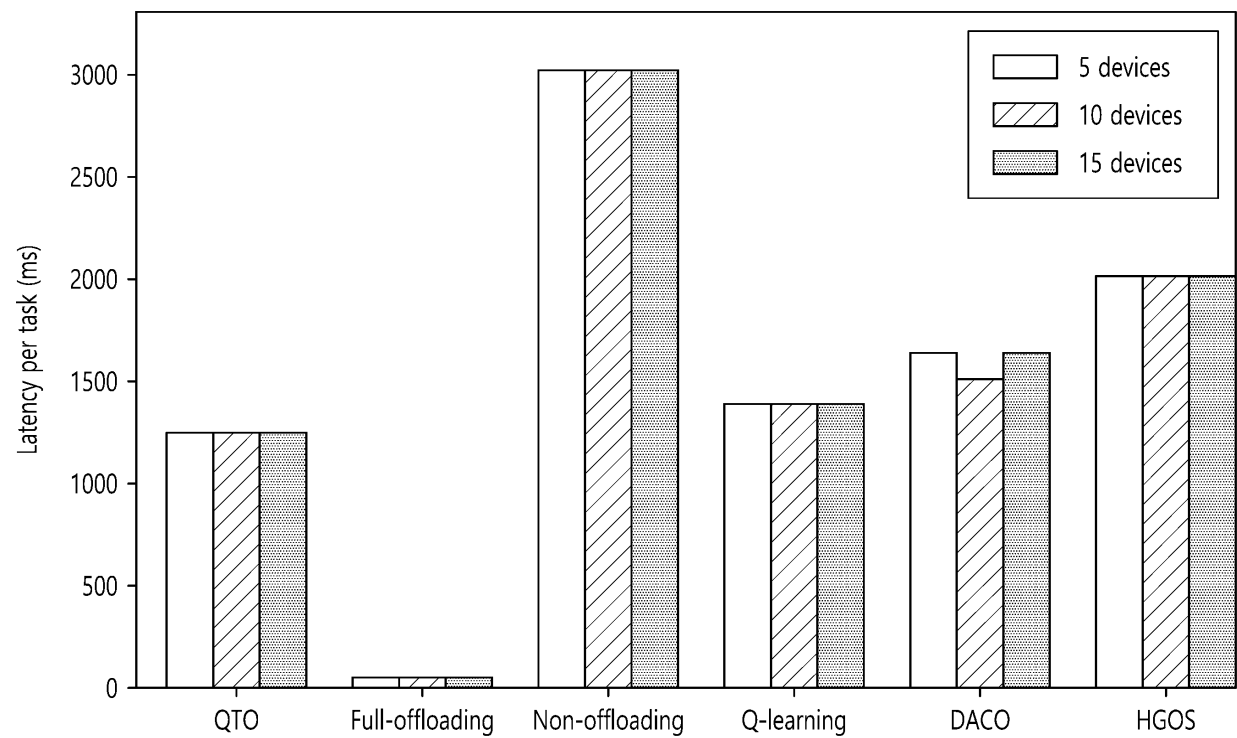
Algorithm : Q-learning Task Offloading Scheme (QTO)

- 1: Initialize $C, f, N, B_1^{(0)}, \dots, B_M^{(0)}, p^{(1)}, b^{(1)}, \alpha, \gamma$ and ε
 - 2: Set $\mathbf{Q} \leftarrow \bar{\mathbf{Q}}$ according to the transfer learning technique
 - 3: **for** $k = 1, 2, 3, \dots$ **do**
 - 4: Estimate the amount of harvested energy $p^{(k)}$
 - 5: Observe the radio transmission rate,
 $B_1^{(k-1)}, \dots, B_M^{(k-1)}$, and the current
battery level $b^{(k)}$
 - 6: $s^{(k)} = [B_1^{(k-1)}, \dots, B_M^{(k-1)}, p^{(k)}, b^{(k)}]$
 - 7: Select $\mathbf{a}^{(k)} = [i^{(k)}, x^{(k)}] \in \mathbf{A}$ via decaying ε -greedy policy
 - 8: Offload $x^{(k)}$ C data to the selected i -th edge node
 - 9: Observe the battery level
 - 10: Evaluate the energy consumption, computation
latency and task drop loss
 - 11: Evaluate $R^{(k)}$ via (7)
 - 12: Update $\mathbf{Q}(s^{(k)}, \mathbf{a}^{(k)})$ via (8)
 - 13: **end for**
-

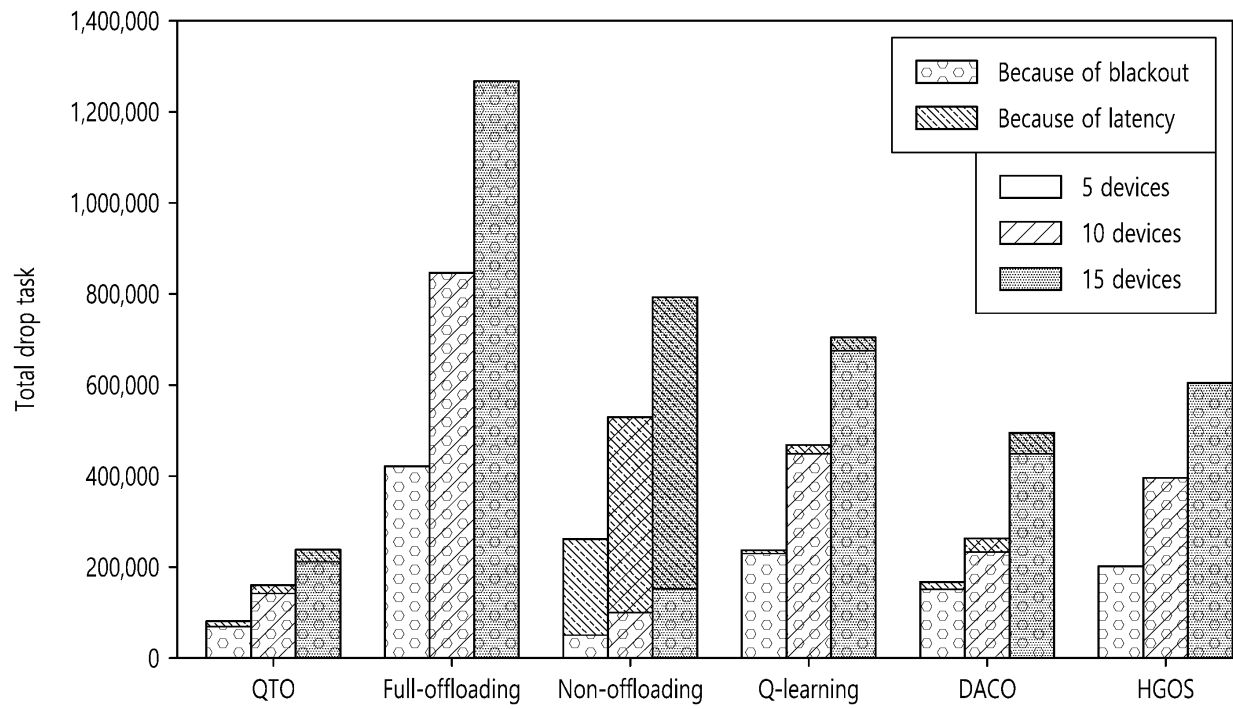
도면5



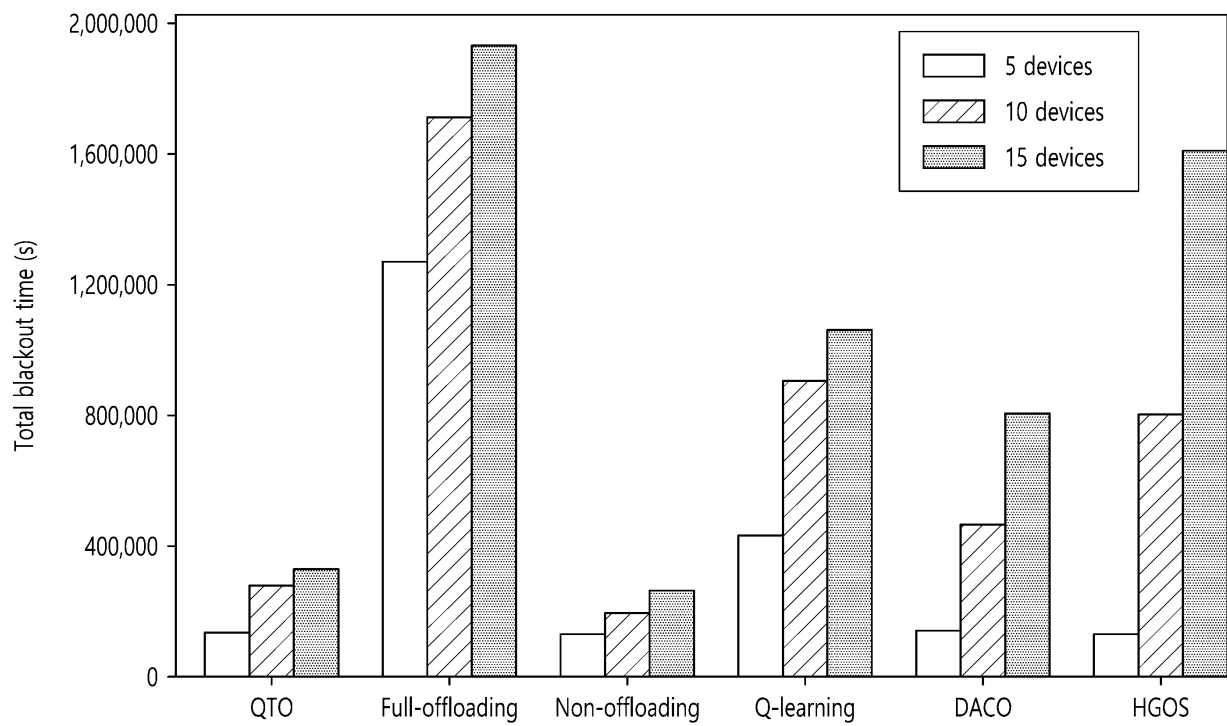
도면6



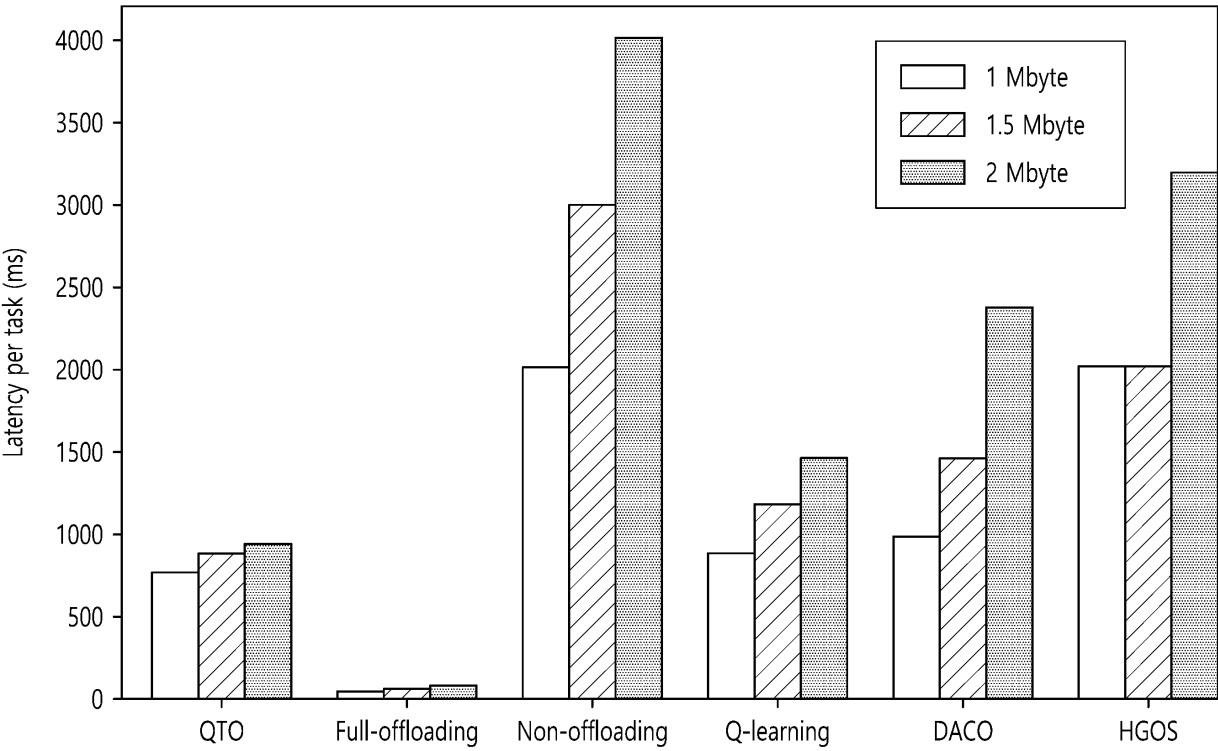
도면7



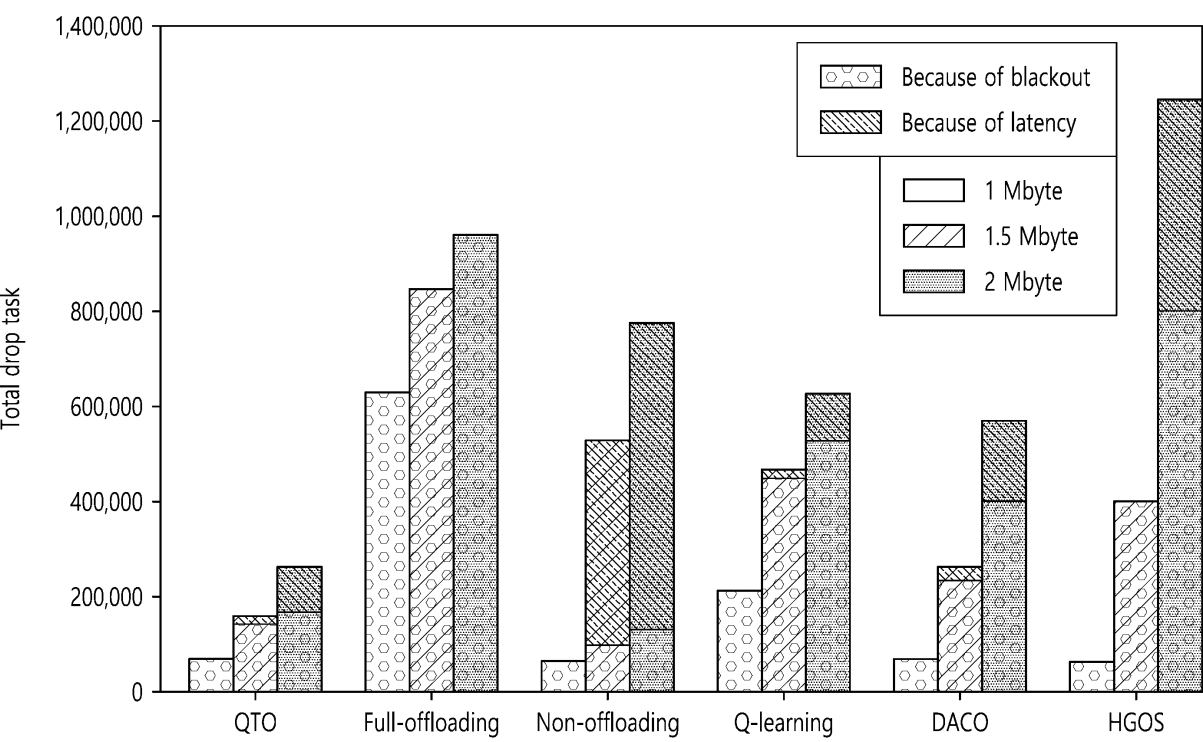
도면8



도면9



도면10



도면11

