



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2023년08월01일

(11) 등록번호 10-2562732

(24) 등록일자 2023년07월28일

(51) 국제특허분류(Int. Cl.)

G06F 9/50 (2018.01)

(52) CPC특허분류

G06F 9/5094 (2013.01)

G06F 9/5077 (2013.01)

(21) 출원번호 10-2021-0115911

(22) 출원일자 2021년08월31일

심사청구일자 2021년08월31일

(65) 공개번호 10-2023-0032754

(43) 공개일자 2023년03월07일

(56) 선행기술조사문헌

CN113220356 A

CN112860429 A

CN112600921 A

KR1020210069588 A

(73) 특허권자

연세대학교 산학협력단

서울특별시 서대문구 연세로 50 (신촌동, 연세대학교)

(72) 발명자

정중문

서울특별시 용산구 이촌로 181, 104동 101호(이촌동, 한강대우아파트)

정동준

서울특별시 서대문구 연희로10길 43-7, 화이트빌 E동 404호(연희동)

김준성

경기도 용인시 수지구 동천로 64, 504동 1401호(동천동, 동천마을동문굿모닝힐5차아파트)

(74) 대리인

민영준

전체 청구항 수 : 총 16 항

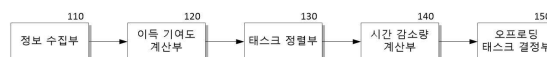
심사관 : 김종기

(54) 발명의 명칭 MEC 기반 무선 네트워크의 태스크 오프로딩 장치 및 방법

(57) 요약

본 발명은 협동 게임 이론에 기반하여 각 태스크에 대한 오프로딩 시의 이득을 계산하여, 오프로딩할 태스크의 우선 순위를 판별하여 정렬하고, 우선 순위 정렬된 다수의 태스크에서 오프로딩될 개수를 에너지 소비량이 최소가 되도록 결정하여 서비스 제한 시간을 만족시키면서 에너지 소비가 최소화되도록 오프로딩을 수행하는 MEC 기반 무선 네트워크의 태스크 오프로딩 장치 및 방법을 제공한다.

대표도 - 도2



(52) CPC특허분류

H04W 4/70 (2018.02)

H04W 88/18 (2019.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	1761002360
과제번호	20017102
부처명	소방청
과제관리(전문)기관명	한국산업기술평가관리원
연구사업명	국민소방협력초기대응현장지원기술개발(R&D)
연구과제명	소방 안전관리자용 화재 초기 대응력 향상 시스템
기 여 율	1/1
과제수행기관명	연세대학교 산학협력단
연구기간	2021.05.01 ~ 2021.12.31

공지예외적용 : 있음

명세서

청구범위

청구항 1

다수의 IoT 기기와 적어도 하나의 사용자 단말 및 적어도 하나의 MEC 서버를 포함하는 MEC 기반 무선 네트워크에서 상기 다수의 IoT 기기에서 발생되어 상기 사용자 단말로 전송된 다수의 태스크 중 상기 MEC 서버로 오프로딩할 태스크를 결정하기 위한 정보를 수집하는 정보 수집부;

협력 게임 이론에 따라 다수의 태스크에서 오프로딩될 태스크의 집합인 협동 연합을 가능한 모든 조합에 따라 구성하는 경우, 협동 연합을 오프로딩하여 발생할 수 있는 에너지 이득인 협력 이득에서 각 태스크의 이득 기여도를 수집된 정보를 이용하여 계산하는 이득 기여도 계산부;

계산된 각 태스크의 이득 기여도에 따라 다수의 태스크를 우선 순위 정렬하는 태스크 정렬부;

계산된 각 태스크의 이득 기여도를 기반으로 각 태스크를 오프로딩 하는 경우, 감소시킬 수 있는 감소 시간을 계산하는 시간 감소량 계산부; 및

기지정된 서비스 제한 시간을 만족하면서 감소 시간을 고려한 상기 다수의 IoT 기기와 상기 사용자 단말의 에너지 소모량의 합으로 계산되는 총 에너지 소비 프로파일이 최소가 되도록 하는 오프로딩 개수를 계산하고, 우선 순위 정렬된 다수의 태스크에서 정렬된 순서에 따라 계산된 오프로딩 개수의 태스크를 오프로딩 태스크로 결정하는 오프로딩 태스크 결정부를 포함하는 태스크 오프로딩 장치.

청구항 2

제1항에 있어서, 상기 이득 기여도 계산부는

상기 이득 기여도를 협력 게임 이론의 샤프리 값(Shapley Value)(ϕ)으로 수학식

$$\varphi_i(v) = \sum_{\mathbb{C} \subseteq \mathbf{N} \setminus \{i\}} \frac{|\mathbb{C}|! (|\mathbf{N}| - |\mathbb{C}| - 1)!}{|\mathbf{N}|!} (v(\mathbb{C} \cup \{i\}) - v(\mathbb{C}))$$

(여기서 $\mathbb{C} \subseteq \mathbf{N} \setminus \{i\}$ 는 협동 연합(\mathbb{C})이 N개의 태스크(I_1, \dots, I_N) 중 제i 태스크(I_i)를 제외한 나머지 태스크의 조합으로 구성되는 모든 태스크 집합임을 나타내고, $v(\mathbb{C})$ 는 협동 연합(\mathbb{C})에 포함된 태스크가 전체 에너지 이득을 향상시키기 위해 협력함으로써 획득할 수 있는 이득을 계산하기 위한 특성 함수로서 협력 이득을 나타낸다.)에 따라 계산하고,

상기 협력 이득($v(\mathbb{C})$)은 상기 협동 연합(\mathbb{C})에 포함되는 태스크의 구성에 따라 상기 협동 연합(\mathbb{C})이 오프로딩되어 상기 MEC 서버에서 처리될 때의 에너지 소비량을 나타내는 협력 에너지 소비량($E_{\mathbb{C}}$)에 기반하여 수학식

$$v(\mathbb{C}) = \gamma \sum_{i \in \mathbb{C}} f_o(i) E_{\mathbb{C}}$$

(여기서 $f_o(i)$ 는 협동 연합(\mathbb{C})에 포함되는 각 태스크(I_i)의 발생 빈도를 나타내고, γ 는 협력에 대한 이익 전환율(payload conversion rate)을 나타내는 밸런스 파라미터(balancing parameter)로서 미리 지정된 상수 값으로 설정될 수 있다.)

에 따라 계산하는 태스크 오프로딩 장치.

청구항 3

제2항에 있어서, 상기 시간 감소량 계산부는

상기 다수의 태스크 각각의 이득 기여도를 상기 사용자 단말의 프로세서가 각 태스크 처리하기 위해 사용하는 프로세싱 전력으로 나누어 상기 감소 시간을 계산하는 태스크 오프로딩 장치.

청구항 4

제3항에 있어서, 상기 오프로딩 태스크 결정부는

상기 오프로딩 개수(n^*)를 수학적식

$$n^* = \left\lfloor \frac{T_{th} - NT_U(I_k)}{T_E(I_k) - T_U(I_k) - T_R(I_k)} \right\rfloor$$

(여기서 $\lfloor \cdot \rfloor$ 는 바닥 함수, T_{th} 는 서비스 제한 시간, N 은 총 태스크 개수, $T_U(I_k)$ 는 로컬 처리 시간, $T_E(I_k)$ 는 오프로딩 처리시간을 나타내고, $T_R(I_k)$ 는 감소 시간을 나타낸다)

에 따라 결정하는 태스크 오프로딩 장치.

청구항 5

제4항에 있어서, 상기 로컬 처리 시간($T_U(I_k)$)은 IoT 기기에서 사용자 단말로의 태스크 전송 시간($T_{I,tr}$)과 사용자 단말의 태스크 처리 시간($T_{U,pr}$)의 합($T_U = T_{I,tr} + T_{U,pr}$)으로 수학적식

$$T_U = \frac{d_k}{r_I} + \frac{W_{k,U}}{f_{max}}$$

(여기서 d_k 는 태스크의 데이터 크기, r_I 는 IoT 기기가 사용자 단말로 태스크(I_k)를 전송하는 태스크 전송 속도, $W_{k,U}$ 는 태스크를 처리하는데 요구되는 사용자 단말의 태스크 처리 사이클 수, f_{max} 는 사용자 단말 프로세서의 최대 주파수)

로 계산되고,

상기 오프로딩 처리 시간(T_E)은 IoT 기기에서 사용자 단말로의 태스크 전송 시간($T_{I,tr}$)과 사용자 단말에서 MEC 서버로의 오프로딩 전송 시간($T_{U,tr}$) 및 MEC 서버의 태스크 처리 시간($T_{E,pr}$)의 합($T_E = T_{I,tr} + T_{U,tr} + T_{E,pr}$)으로 수학적식

$$T_E = \frac{d_k}{r_I} + \frac{d_k}{r_U} + \frac{W_{k,E}}{f_E}$$

(여기서 r_U 는 사용자 단말이 MEC 서버로 태스크(I_k)를 전송하는 오프로딩 전송 속도, $W_{k,E}$ 는 태스크를 처리하는데 요구되는 MEC 서버의 태스크 처리 사이클 수, f_E 는 MEC 서버 프로세서의 주파수)

으로 계산되는 태스크 오프로딩 장치.

청구항 6

제2항에 있어서, 상기 이득 기여도 계산부는

상기 협력 에너지 소비량(E_C)을 MEC 서버에서 처리되는 각 태스크에 대한 처리 에너지의 합으로 수학적식

$$E_C = \sum_{k=1}^n P_{pr} T_{E,pr}$$

(여기서 n 은 협동 연합(C)에 포함되는 태스크 개수, P_{pr} 은 MEC 서버가 태스크를 처리하기 위해 사용하는 프로세싱 전력, $T_{E,pr}$ 은 MEC 서버의 태스크 처리 시간)

에 따라 계산하는 태스크 오프로딩 장치.

청구항 7

제6항에 있어서, 상기 IoT 기기의 에너지 소비량(E_i)은 상기 다수의 태스크(I_1, \dots, I_N)에 대한 사용자 단말(UE)로의 전송 에너지, 상기 다수의 태스크(I_1, \dots, I_N) 중 오프로딩되는 n 개의 태스크(I_1, \dots, I_n)가 사용자 단말(UE)을 통해 MEC 서버로 오프로딩 처리되는 동안의 유향 에너지 및 나머지 태스크(I_{n+1}, \dots, I_N)가 사용자 단말(UE)에서 로컬 처리되는 동안의 유향 에너지의 합으로 계산되는 태스크 오프로딩 장치.

청구항 8

제7항에 있어서, 상기 사용자 단말(UE)의 에너지 소비량(E_U)은 상기 다수의 태스크(I_1, \dots, I_N)가 전송되는 동안의 유향 에너지, 상기 다수의 태스크(I_1, \dots, I_N) 중 오프로딩되는 n 개의 태스크(I_1, \dots, I_n)를 MEC 서버로 전송하는 전송 에너지와 오프로딩된 n 개의 태스크(I_1, \dots, I_n)가 MEC 서버에서 처리되는 동안의 유향 에너지 및 사용자 단말(UE)이 나머지 태스크(I_{n+1}, \dots, I_N)를 로컬 처리하는 프로세싱 에너지의 합으로 계산되는 태스크 오프로딩 장치.

청구항 9

다수의 IoT 기기와 적어도 하나의 사용자 단말 및 적어도 하나의 MEC 서버를 포함하는 MEC 기반 무선 네트워크에서 상기 다수의 IoT 기기에서 발생되어 상기 사용자 단말로 전송된 다수의 태스크 중 상기 MEC 서버로 오프로딩할 태스크를 결정하는 태스크 오프로딩 장치의 태스크 오프로딩 방법에 있어서,

오프로딩될 태스크를 선택하기 위한 정보를 수집하는 단계;

수집된 정보를 이용하여 협력 게임 이론에 따라 다수의 태스크에서 오프로딩될 태스크의 집합인 협동 연합을 가능한 모든 조합에 따라 구성하는 경우, 협동 연합을 오프로딩하여 발생할 수 있는 에너지 이득인 협력 이득에서 각 태스크의 이득 기여도를 계산하는 단계;

계산된 각 태스크의 이득 기여도에 따라 다수의 태스크를 우선 순위 정렬하는 단계;

계산된 각 태스크의 이득 기여도를 기반으로 각 태스크를 오프로딩 하는 경우, 감소시킬 수 있는 감소 시간을 계산하는 단계; 및

기지정된 서비스 제한 시간을 만족하면서 감소 시간을 고려한 상기 다수의 IoT 기기와 상기 사용자 단말의 에너지 소비량의 합으로 계산되는 총 에너지 소비 프로파일의 최소가 되도록 하는 오프로딩 개수를 계산하고, 우선 순위 정렬된 다수의 태스크에서 정렬된 순서에 따라 계산된 오프로딩 개수의 태스크를 오프로딩 태스크로 결정하는 단계를 포함하는 태스크 오프로딩 방법.

청구항 10

제9항에 있어서, 상기 태스크의 이득 기여도를 계산하는 단계는

상기 이득 기여도를 협력 게임 이론의 샤플리 값(Shapley Value)(ϕ)으로 수학식

$$\varphi_i(v) = \sum_{C \subseteq N \setminus \{i\}} \frac{|C|! (|N| - |C| - 1)!}{|N|!} (v(C \cup \{i\}) - v(C))$$

(여기서 $C \subseteq N \setminus \{i\}$ 는 협동 연합(C)이 N 개의 태스크(I_1, \dots, I_N) 중 제 i 태스크(I_i)를 제외한 나머지 태스크의 조합으로 구성되는 모든 태스크 집합임을 나타내고, $v(C)$ 는 협동 연합(C)에 포함된 태스크가 전체 에너지를 향상시키기 위해 협력함으로써 획득할 수 있는 이득을 계산하기 위한 특성 함수로서 협력 이득을 나타낸다.)

에 따라 계산하고,

상기 협력 이득($v(C)$)은 상기 협동 연합(C)에 포함되는 태스크의 구성에 따라 상기 협동 연합(C)이 오프로딩

되어 상기 MEC 서버에서 처리될 때의 에너지 소비량을 나타내는 협력 에너지 소비량($E_{\mathbb{C}}$)에 수학적

$$v(\mathbb{C}) = \gamma \sum_{i \in \mathbb{C}} f_o(i) E_{\mathbb{C}}$$

(여기서 $f_o(i)$ 는 협동 연합(\mathbb{C})에 포함되는 각 태스크(I_i)의 발생 빈도를 나타내고, γ 는 협력에 대한 이익 전환율(payload conversion rate)을 나타내는 밸런스 파라미터(balancing parameter)로서 미리 지정된 상수 값으로 설정될 수 있다.)

에 따라 계산하는 태스크 오프로딩 방법.

청구항 11

제10항에 있어서, 상기 감소 시간을 계산하는 단계는

상기 다수의 태스크 각각의 이득 기여도를 상기 사용자 단말의 프로세서가 각 태스크 처리하기 위해 사용하는 프로세싱 전력으로 나누어 상기 감소 시간을 계산하는 태스크 오프로딩 방법.

청구항 12

제11항에 있어서, 상기 오프로딩 태스크로 결정하는 단계는

상기 오프로딩 개수(n^*)를 수학적

$$n^* = \left\lfloor \frac{T_{th} - NT_U(I_k)}{T_E(I_k) - T_U(I_k) - T_R(I_k)} \right\rfloor$$

(여기서 $\lfloor \cdot \rfloor$ 는 바닥 함수, T_{th} 는 서비스 제한 시간, N 은 총 태스크 개수, $T_U(I_k)$ 는 로컬 처리 시간, $T_E(I_k)$ 는 오프로딩 처리시간을 나타내고, $T_R(I_k)$ 는 감소 시간을 나타낸다)

에 따라 결정하는 태스크 오프로딩 방법.

청구항 13

제12항에 있어서, 상기 로컬 처리 시간($T_U(I_k)$)은 IoT 기기에서 사용자 단말로 태스크 전송 시간($T_{I,tr}$)과 사용자 단말의 태스크 처리 시간($T_{U,pr}$)의 합($T_U = T_{I,tr} + T_{U,pr}$)으로 수학적

$$T_U = \frac{d_k}{r_I} + \frac{W_{k,U}}{f_{max}}$$

(여기서 d_k 는 태스크의 데이터 크기, r_I 는 IoT 기기가 사용자 단말로 태스크(I_k)를 전송하는 태스크 전송 속도, $W_{k,U}$ 는 태스크를 처리하는데 요구되는 사용자 단말의 태스크 처리 사이클 수, f_{max} 는 사용자 단말 프로세서의 최대 주파수)

로 계산되고,

상기 오프로딩 처리 시간(T_E)은 IoT 기기에서 사용자 단말로 태스크 전송 시간($T_{I,tr}$)과 사용자 단말에서 MEC 서버로의 오프로딩 전송 시간($T_{U,tr}$) 및 MEC 서버의 태스크 처리 시간($T_{E,pr}$)의 합($T_E = T_{I,tr} + T_{U,tr} + T_{E,pr}$)으로 수학적

$$T_E = \frac{d_k}{r_I} + \frac{d_k}{r_U} + \frac{W_{k,E}}{f_E}$$

(여기서 r_U 는 사용자 단말이 MEC 서버로 태스크(I_k)를 전송하는 오프로딩 전송 속도, $W_{k,E}$ 는 태스크를 처리하는

데 요구되는 MEC 서버의 태스크 처리 사이클 수, f_e 는 MEC 서버 프로세서의 주파수)

으로 계산되는 태스크 오프로딩 방법.

청구항 14

제10항에 있어서, 상기 이득 기여도를 계산하는 단계는

상기 협력 에너지 소비량(E_C)을 MEC 서버에서 처리되는 각 태스크에 대한 처리 에너지의 합으로 수학식

$$E_C = \sum_{k=1}^n P_{pr} T_{E,pr}$$

(여기서 n 은 협동 연합(\mathbb{C})에 포함되는 태스크 개수, P_{pr} 은 MEC 서버가 태스크를 처리하기 위해 사용하는 프로세싱 전력, $T_{E,pr}$ 은 MEC 서버의 태스크 처리 시간)

에 따라 계산하는 태스크 오프로딩 방법.

청구항 15

제14항에 있어서, 상기 IoT 기기의 에너지 소비량(E_i)은 상기 다수의 태스크(I_1, \dots, I_N)에 대한 사용자 단말(UE)로의 전송 에너지, 상기 다수의 태스크(I_1, \dots, I_N) 중 오프로딩되는 n 개의 태스크(I_1, \dots, I_n)가 사용자 단말(UE)을 통해 MEC 서버로 오프로딩 처리되는 동안의 유휴 에너지 및 나머지 태스크(I_{n+1}, \dots, I_N)가 사용자 단말(UE)에서 로컬 처리되는 동안의 유휴 에너지의 합으로 계산되는 태스크 오프로딩 방법.

청구항 16

제15항에 있어서, 상기 사용자 단말(UE)의 에너지 소비량(E_u)은 상기 다수의 태스크(I_1, \dots, I_N)가 전송되는 동안의 유휴 에너지, 상기 다수의 태스크(I_1, \dots, I_N) 중 오프로딩되는 n 개의 태스크(I_1, \dots, I_n)를 MEC 서버로 전송하는 전송 에너지와 오프로딩된 n 개의 태스크(I_1, \dots, I_n)가 MEC 서버에서 처리되는 동안의 유휴 에너지 및 사용자 단말(UE)이 나머지 태스크(I_{n+1}, \dots, I_N)를 로컬 처리하는 프로세싱 에너지의 합으로 계산되는 태스크 오프로딩 방법.

발명의 설명

기술 분야

[0001] 본 발명은 태스크 오프로딩 장치 및 방법에 관한 것으로, MEC 기반 무선 네트워크에서 협력을 통해 서비스 시간 제약 하에서 에너지를 최소화할 수 있도록 태스크 오프로딩을 수행하는 태스크 오프로딩 장치 및 방법에 관한 것이다.

배경 기술

[0002] 사물 인터넷(Internet of Things: 이하 IoT) 기술의 향상으로 인해 사용자는 모바일 사용자 단말(User Equipment: UE)을 이용하여 시스템(예: 스마트 빌딩 관리 시스템, 스마트 팩토리 등)에 통합된 수많은 IoT 기기를 제어할 수 있다. 다만 제어 가능한 IoT 기기의 수가 증가함에 따라 처리해야하는 데이터의 양과 그에 따른 에너지 소비가 함께 증가했다. 그러나 많은 IoT 기기와 대부분의 사용자 단말 배터리로 작동하므로 에너지 소비를 최소화하는 것이 매우 중요하다. 또한 증강 현실(AR), 제스처 및 얼굴 인식, 음성 제어, 스마트 홈, 스마트 농업 및 스마트 모니터링과 같은 많은 IoT 애플리케이션은 대기 시간에 매우 민감하므로, 서비스 제한 시간 내에 태스크를 처리해야 한다.

[0003] 이를 위한 해결책 중 하나는 사용자 단말이 네트워크의 엣지에 분산된 다수의 엣지 서버로 태스크를 전송하여 처리함으로써 사용자 단말의 하드웨어 성능 제한과 에너지 소모를 저감시키는 다중 액세스 엣지 컴퓨팅(Multi-access Edge Computing: 이하 MEC) 시스템이 IoT 기기 및 UE 대신 계산을 수행하여 에너지를 절약하도록 하는 것이다. MEC 시스템에서는 사용자 단말이 처리되어야 하는 태스크를 가까운 엣지에 위치하는 MEC 서버로 오프로딩하여 분산 처리할 수 있도록 함으로써 태스크 전송에 따른 지연 시간을 크게 단축하여 서비스 제한 시간 내

에 태스크를 처리할 수 있도록 한다.

[0004] 이때 사용자 단말은 서비스 제한 시간이 있는 다수의 태스크에 대해 제한된 서비스 시간을 만족하면서 동시에 다수의 IoT 기기와 사용자 단말의 에너지 소비를 최소화할 수 있는 최적의 태스크 오프로딩을 수행할 수 있어야 한다.

선행기술문헌

특허문헌

[0005] (특허문헌 0001) 한국 공개 특허 제10-2021-0067468호 (2021.06.08 공개)

발명의 내용

해결하려는 과제

[0006] 본 발명의 목적은 서비스 제한 시간을 만족하면서 에너지 소비가 최소화되도록 태스크 오프로딩을 수행할 수 있는 태스크 오프로딩 장치 및 방법을 제공하는데 있다.

[0007] 본 발명의 다른 목적은 협동 게임 이론에 기반하여 각 태스크의 오프로딩 이득을 계산하여, 오프로딩할 태스크와 개수를 선택함으로써 에너지 효율적 태스크 오프로딩을 수행할 수 있는 태스크 오프로딩 장치 및 방법을 제공하는데 있다.

과제의 해결 수단

[0008] 상기 목적을 달성하기 위한 본 발명의 일 실시예에 따른 태스크 오프로딩 장치는 다수의 IoT 기기와 적어도 하나의 사용자 단말 및 적어도 하나의 MEC 서버를 포함하는 MEC 기반 무선 네트워크에서 상기 다수의 IoT 기기에서 발생되어 상기 사용자 단말로 전송된 다수의 태스크 중 상기 MEC 서버로 오프로딩할 태스크를 결정하기 위한 정보를 수집하는 정보 수집부; 협력 게임 이론에 따라 다수의 태스크에서 오프로딩될 태스크의 집합인 협동 연합을 가능한 모든 조합에 따라 구성하는 경우, 협동 연합을 오프로딩하여 발생할 수 있는 에너지 이득인 협력 이득에서 각 태스크의 이득 기여도를 수집된 정보를 이용하여 계산하는 이득 기여도 계산부; 계산된 각 태스크의 이득 기여도에 따라 다수의 태스크를 우선 순위 정렬하는 태스크 정렬부; 계산된 각 태스크의 이득 기여도를 기반으로 각 태스크(I_k)를 오프로딩 하는 경우, 감소시킬 수 있는 감소 시간을 계산하는 시간 감소량 계산부; 및 기지정된 서비스 제한 시간을 만족하면서 감소 시간을 고려한 상기 다수의 IoT 기기와 상기 사용자 단말의 에너지 소모량의 합으로 계산되는 총 에너지 소비 프로파일의 최소가 되도록 하는 오프로딩 개수를 계산하고, 우선 순위 정렬된 다수의 태스크에서 정렬된 순서에 따라 계산된 오프로딩 개수의 태스크를 오프로딩 태스크로 결정하는 오프로딩 태스크 결정부를 포함한다.

[0009] 상기 이득 기여도 계산부는 상기 이득 기여도를 협력 게임 이론의 샵플리 값(Shapley Value)(ϕ)으로 수학식

$$\varphi_i(v) = \sum_{C \subseteq N \setminus \{i\}} \frac{|C|! (|N| - |C| - 1)!}{|N|!} (v(C \cup \{i\}) - v(C))$$

[0010]

[0011] (여기서 $C \subseteq N \setminus \{i\}$ 는 협동 연합(C)이 N 개의 태스크(I_1, \dots, I_N) 중 제 i 태스크(I_i)를 제외한 나머지 태스크의 조합으로 구성되는 모든 태스크 집합임을 나타내고, $v(C)$ 는 협동 연합(C)에 포함된 태스크가 전체 에너지 이득을 향상시키기 위해 협력함으로써 획득할 수 있는 이득을 계산하기 위한 특성 함수로서 협력 이득을 나타낸다.)에 따라 계산하고, 상기 협력 이득($v(C)$)은 상기 협동 연합(C)에 포함되는 태스크의 구성에 따라 상기 협동 연합(C)이 오프로딩되어 상기 MEC 서버에서 처리될 때의 에너지 소비량을 나타내는 협력 에너지 소비량(E_C)에 기반하여 수학식

$$v(\mathbb{C}) = \gamma \sum_{i \in \mathbb{C}} f_o(i) E_{\mathbb{C}}$$

[0012]

[0013] (여기서 $f_o(i)$ 는 협동 연합(\mathbb{C})에 포함되는 각 태스크(I_i)의 발생 빈도를 나타내고, γ 는 협력에 대한 이익 전환율(payload conversion rate)을 나타내는 밸런싱 파라미터(balancing parameter)로서 미리 지정된 상수 값으로 설정될 수 있다.)에 따라 계산할 수 있다.

[0014] 상기 시간 감소량 계산부는 상기 다수의 태스크 각각의 이득 기여도를 상기 사용자 단말의 프로세서가 각 태스크 처리하기 위해 사용하는 프로세싱 전력으로 나누어 상기 감소 시간을 계산할 수 있다.

[0015] 상기 오프로딩 태스크 결정부는 상기 오프로딩될 태스크 개수(n^*)를 수학식

$$n^* = \left\lfloor \frac{T_{th} - NT_U(I_k)}{T_E(I_k) - T_U(I_k) - T_R(I_k)} \right\rfloor$$

[0016]

[0017] (여기서 T_{th} 는 서비스 제한 시간, N 은 총 태스크 개수, $T_U(I_k)$ 는 로컬 처리 시간, $T_E(I_k)$ 는 오프로딩 처리시간을 나타내고, $T_R(I_k)$ 는 감소 시간을 나타낸다)에 따라 결정할 수 있다.

[0018] 상기 목적을 달성하기 위한 본 발명의 다른 실시예에 따른 태스크 오프로딩 방법은 오프로딩될 태스크를 선택하기 위한 정보를 수집하는 단계; 수집된 정보를 이용하여 협력 게임 이론에 따라 다수의 태스크에서 오프로딩될 태스크의 집합인 협동 연합을 가능한 모든 조합에 따라 구성하는 경우, 협동 연합을 오프로딩하여 발생할 수 있는 에너지 이득인 협력 이득에서 각 태스크의 이득 기여도를 계산하는 단계; 계산된 각 태스크의 이득 기여도에 따라 다수의 태스크를 우선 순위 정렬하는 단계; 계산된 각 태스크의 이득 기여도를 기반으로 각 태스크(I_k)를 오프로딩 하는 경우, 감소시킬 수 있는 감소 시간을 계산하는 단계; 및 기지정된 서비스 제한 시간을 만족하면서 감소 시간을 고려한 상기 다수의 IoT 기기와 상기 사용자 단말의 에너지 소모량의 합으로 계산되는 총 에너지 소비 프로파일이 최소가 되도록 하는 오프로딩 개수를 계산하고, 우선 순위 정렬된 다수의 태스크에서 정렬된 순서에 따라 계산된 오프로딩 개수의 태스크를 오프로딩 태스크로 결정하는 단계를 포함한다.

발명의 효과

[0019] 따라서, 본 발명의 실시예에 따른 태스크 오프로딩 장치 및 방법은 협동 게임 이론에 기반하여 각 태스크에 대한 오프로딩 시의 이득을 계산하여, 오프로딩할 태스크의 우선 순위를 판별하여 정렬하고, 우선 순위 정렬된 다수의 태스크에서 오프로딩될 개수를 에너지 소비량이 최소가 되도록 결정할 수 있다. 그러므로 서비스 제한 시간을 만족시키면서 에너지 소비가 최소화되도록 오프로딩을 수행할 수 있다.

도면의 간단한 설명

[0020] 도 1은 MEC 기반 네트워크 오프로딩 시스템의 일 예를 나타낸다.

도 2는 본 발명의 일 실시예에 따른 태스크 오프로딩 장치의 개략적 구조를 나타낸다.

도 3은 본 발명의 일 실시예에 따른 태스크 오프로딩 방법을 나타낸다.

발명을 실시하기 위한 구체적인 내용

[0021] 본 발명과 본 발명의 동작상의 이점 및 본 발명의 실시예에 의하여 달성되는 목적을 충분히 이해하기 위해서는 본 발명의 바람직한 실시예를 예시하는 첨부 도면 및 첨부 도면에 기재된 내용을 참조하여야만 한다.

[0022] 이하, 첨부한 도면을 참조하여 본 발명의 바람직한 실시예를 설명함으로써, 본 발명을 상세히 설명한다. 그러나, 본 발명은 여러 가지 상이한 형태로 구현될 수 있으며, 설명하는 실시예에 한정되는 것이 아니다. 그리고, 본 발명을 명확하게 설명하기 위하여 설명과 관계없는 부분은 생략되며, 도면의 동일한 참조부호는 동일한 부재임을 나타낸다.

[0023] 명세서 전체에서, 어떤 부분이 어떤 구성요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라, 다른 구성요소를 더 포함할 수 있는 것을 의미한다. 또한, 명세서에 기재된 "...부", "...기", "모듈", "블록" 등의 용어는 적어도 하나의 기능이나 동작을 처리하는 단위를

의미하며, 이는 하드웨어나 소프트웨어 또는 하드웨어 및 소프트웨어의 결합으로 구현될 수 있다.

[0024] 도 1은 MEC 기반 네트워크 오프로딩 시스템의 일 예를 나타낸다.

[0025] 도 1을 참조하면, 본 실시예에 따른 MEC 기반 네트워크 오프로딩 시스템에서는 적어도 하나의 MEC 서버(MEC Server)와 적어도 하나의 MEC 서버로 태스크 오프로딩을 수행할 수 있는 적어도 하나의 사용자 단말(UE) 및 적어도 하나의 사용자 단말(UE)에 대응하여 클러스터링된 다수의 IoT 기기(IoT Device)를 포함한다. 여기서 MEC 서버는 사용자 단말(UE)에 가장 인접하여 위치하는 기지국(Base Station)에 포함될 수 있다. 그리고 하나의 MEC 서버와 적어도 하나의 사용자 단말(UE) 각각은 프로세서와 메모리를 포함하여 전송된 태스크를 처리할 수 있다.

[0026] 우선 사용자 단말(UE)에 대응하여 클러스터링된 다수의 IoT 기기는 대응하는 사용자 단말(UE)로 태스크(I_k , 여기서 $k \in N = \{1, \dots, N\}$)가 처리되도록 요청할 수 있다. 즉 다수의 IoT 기기는 각각 사용자 단말(UE)로 처리되어야 할 태스크(I_k)를 전송할 수 있다. 사용자 단말(UE)은 대응하는 다수의 IoT 기기에서 전송된 태스크(I_k)를 처리한다. 사용자 단말(UE)은 다수의 IoT 기기에서 전송된 태스크(I_k)를 로컬 프로세서를 이용하여 로컬 처리할 수도 있으나, 사용자 단말(UE)의 에너지 소모를 저감하고, 각 태스크(I_k)에 지정된 서비스 제한 시간을 만족시킬 수 있도록 일부 태스크(I_k)를 MEC 서버로 오프로딩하여 MEC 서버에서 해당 태스크(I_k)가 처리되도록 할 수 있다. 이때 사용자 단말(UE)은 본 실시예에 따른 오프로딩 장치(미도시)를 포함하여 다수의 IoT 기기에서 전송된 다수의 태스크 중 MEC 서버로 오프로딩할 태스크를 선택할 수 있다.

[0027] 본 실시예에서 오프로딩 장치는 대응하는 다수의 IoT 기기에서 전송된 N 개의 전체 태스크($I_1 \sim I_N$)에서 오프로딩될 태스크의 개수(n)와 오프로딩될 태스크를 협동 게임 이론에 기반하여 시간 및 에너지 관점에서 이득이 최대가 되도록 결정한다. 이때 오프로딩 장치는 일 예로 도 1의 좌상단에 나타난 표와 같이, 협동 게임 이론에서 플레이어들이 연합하여 협동함으로써 획득되는 추가 이득에 대한 각 플레이어의 기여도를 나타내는 샵리 값(Shapley Value)(ϕ_k)을 기반으로 오프로딩될 태스크와 로컬 프로세서에서 처리될 태스크를 구분하여 선택할 수 있다. 오프로딩 장치는 태스크 오프로딩 시에 발생하는 이득에서 각 태스크의 이득 기여도를 샵리 값(ϕ_k)에 따라 계산하여 오프로딩될 태스크의 우선 순위를 결정하여, N 개의 전체 태스크($I_1 \sim I_N$) 각각을 샵리 값(ϕ_k)에 기반하여 우선 순위 정렬하고, 우선 순위 정렬된 N 개의 태스크($I_1 \sim I_N$) 중 MEC 서버로 오프로딩될 n 개의 태스크($I_1 \sim I_n$)를 결정할 수 있다.

[0028] 도 2는 본 발명의 일 실시예에 따른 태스크 오프로딩 장치의 개략적 구조를 나타낸다.

[0029] 도 2를 참조하면, 사용자 단말(UE)에 포함되는 오프로딩 장치는 정보 수집부(110), 이득 기여도 계산부(120), 태스크 정렬부(130), 시간 감소량 계산부(140) 및 오프로딩 결정부(150)를 포함할 수 있다.

[0030] 정보 수집부(110)는 대응하는 다수의 IoT 기기에서 전송되는 태스크에 대한 태스크 정보와 IoT 기기와 사용자 단말(UE) 및 사용자 단말(UE)과 MEC 서버 사이의 통신 정보 및 사용자 단말(UE)과 MEC 서버의 연산 능력 정보를 획득한다.

[0031] 정보 수집부(110)는 태스크 정보로 다수의 IoT 기기에서 처리를 요청하는 전체 태스크 개수(N)와 각 태스크(I_k)의 데이터 크기(d_k) 및 서비스 제한 시간(T_{th}) 등을 획득하고, 통신 정보로서 IoT 기기와 사용자 단말(UE)의 각각의 채널 이득(h_l, h_u), 채널 대역폭(B_l, B_u)과 채널의 잡음 전력 밀도($N_{0,l}, N_{0,u}$) 및 전송 전력(P_l, P_u) 등을 획득할 수 있으며, 연산 능력 정보로서 사용자 단말(UE)의 로컬 프로세서의 동작 주파수(f)와 MEC 서버의 프로세서 동작 주파수(f_e) 등을 획득한다.

[0032] 정보 수집부(110)에서 수집된 태스크 정보와 통신 정보 및 연산 능력 정보는 이후 다수의 IoT 기기에서 전송된 태스크를 오프로딩할 때 추가적으로 획득되는 이득 기여도와 시간 감소량에 기반하여 오프로딩될 태스크를 선택하기 위한 연산에 이용된다. 구체적으로 오프로딩될 태스크를 선택하기 위해서는 각 태스크에 대한 로컬 처리와 오프로딩 처리에 따른 총 에너지 소비와 서비스 시간을 계산해야 하며, 총 에너지 소비와 서비스 시간을 계산하기 위해 사용자 단말(UE) 또는 MEC 서버가 각 태스크를 처리하기 위해 요구되는 태스크 처리 사이클 수와 태스크를 전송하기 위해 요구되는 태스크 전송 속도가 먼저 계산되어야 한다.

[0033] 다수의 IoT 기기에서 처리 요청된 N 개의 태스크($I_1 \sim I_N$) 각각을 처리하기 위해 요구되는 태스크 처리 사이클 수

(W_k)는 태스크 정보와 연산 능력 정보를 기반으로 계산될 수 있다.

[0034] 본 실시예에서 IoT 기기가 사용자 단말(UE)로 처리를 요청하는 태스크(I_k)가 발생하는 태스크 발생 빈도($f_o(k)$)는 웹 캐싱의 분포를 나타내는 Zipf 분포를 따르는 것으로 가정한다. 따라서 각 태스크에 대한 태스크 발생 빈도($f_o(k)$)는 Zipf 분포에 기초한 확률 모델에 따라 $f_o(k) = Pr[I_k] = \frac{\Omega}{k^s}$ 로 계산되며, 여기서 Ω 는 $\Omega = \frac{1}{\sum_{k=1}^N k^{-s}}$ 로 계산될 수 있으며 s 는 이론상의 Zipf 분포와의 유사도를 의미하며 일 예로 0.69이다.

[0035] 이때 N 개의 태스크($I_1 \sim I_N$)는 태스크 발생 빈도($f_o(k)$)에 기반하여 미리 정렬될 수 있다.

[0036] 그리고 정보 수집부(110)에서 수집된 태스크(I_k)의 데이터 크기가 d_k 일 경우, 사용자 단말(UE) 또는 MEC의 프로세서가 태스크(I_k)를 처리하기 위해 요구되는 태스크 처리 사이클 수(W_k)는 $W_k = d_k X_k$ 로 계산될 수 있다. 여기서 X_k 는 사용자 단말(UE) 또는 MEC의 프로세서의 비트당 프로세서 사이클이 계산에 사용되는 처리 밀도를 나타내며, IoT 태스크의 특성(예를 들면 계산 복잡도, IoT 태스크 복잡도 등)에 경험적 분포를 갖는 랜덤 변수로 모델링될 수 있다. 따라서 각 태스크(I_k)에 대한 경험적 분포에 따른 확률 밀도 함수(Probability Density

Function: PDF)는 $p_{X_k}(x; \alpha, \beta) = \frac{1}{\beta \Gamma(\alpha)} \left(\frac{x}{\beta}\right)^{\alpha-1} e^{-\frac{x}{\beta}}$, for $x > 0$ 로 나타낼 수 있다. 여기서 $\Gamma()$ 는 감마 분포 함수이고, α 와 β 는 각각 비트당 프로세서 사이클 수의 분포로 사용되는 감마 분포의 모양과 스케일을 나타내며, 일 예로 각각 4 및 200으로 설정될 수 있다.

[0037] 이에 태스크 처리 사이클 수(W_k)의 누적 분포 함수(Cumulative Distribution Function: CDF)는 $F_{W_k}(w) = F_{X_k}(w/d_k)$ 와 같이 유도될 수 있다. 여기서 $F_{X_k}(x) = Pr[X_k \leq x]$ 이다. 그리고 태스크가 서비스 제한 시간(T_{th})을 만족하며 처리될 확률을 나타내는 작업 완료 확률(Work Completion Probability: WCP)(ρ)이 지정되면, 태스크 처리 사이클 수(W_k)의 누적 분포 함수(F_{W_k})는 $F_{W_k}(W_{k,\rho}) \geq \rho$ 를 만족해야 한다. 따라서 태스크 처리 사이클 수(W_k)는 작업 완료 확률(ρ)에 따른 누적 분포 함수(F_{W_k})의 역함수($F_{W_k}^{-1}(\rho)$)로 표현되어, 수학식 1과 같이 계산될 수 있다.

수학식 1

$$W_{k,\rho} = F_{W_k}^{-1}(\rho) = d_k F_{X_k}^{-1}(\rho)$$

[0038]

[0039] 한편, 획득된 통신 정보로부터 다수의 IoT 기기가 사용자 단말(UE)로 태스크(I_k)를 전송하는 태스크 전송 속도(r_l)와 사용자 단말(UE)이 MEC 서버로 태스크(I_k)를 오프로딩 전송 속도(r_u)를 계산할 수 있다.

[0040] 태스크(I_k)의 전송 속도는 서비스 제한 시간을 만족시키기 위해 반드시 고려되어야 사항으로 상기한 바와 같이, 다수의 IoT 기기에서 발생한 태스크(I_k)는 대응하는 사용자 단말(UE)로 전송되고, 사용자 단말(UE)로 전송된 태스크(I_k)는 오프로딩 여부에 따라 MEC 서버로 오프로딩 되어 처리되거나, 사용자 단말(UE) 자체에서 로컬 처리될 수 있다.

[0041] 우선 다수의 IoT 기기가 사용자 단말(UE)로 태스크(I_k)를 전송하는 태스크 전송 속도(r_l)를 계산한다. 사용자 단말(UE)이 M 개의 안테나를 구비하여 통신을 수행한다고 가정하고, 태스크(I_k)는 $\mathbb{E}[|x_{k,I}|^2] = P_I$ (여기서 \mathbb{E} 는 에너지, P_I 는 IoT 기기의 태스크 전송 전력)를 만족하는 원형 대칭 복소 가우스 신호($x_{k,I} \sim \mathcal{CN}(0, P_I)$)

의 형태로 전송되는 것으로 가정한다. 사용자 단말(UE)의 m 번째 안테나에서 수신된 신호는 $y_{m,I} = h_{m,I}x_{k,I} + \sigma_{m,I}$, $m \in \mathbf{M} = \{1, 2, \dots, M\}$ (여기서 $h_{m,I}$ 는 레일리 페이딩 블록 업링크 채널(Rayleigh fading block uplink channel)($h_I \subset \mathbb{C}^{M \times 1}$)에서 m 번째 안테나의 복소 채널 이득을 나타내고, $\sigma_{m,I}$ 는 가산 가우시안 잡음(additive Gaussian noise)을 나타낸다.

[0042] 이에 태스크 전송 속도(r_I)를 수학적 2에 따라 계산할 수 있다.

수학적 2

$$r_I = B_I \log_2 \left(1 + \frac{P_I \|h_I\|^2}{N_{0,I} B_I} \right)$$

[0043]

[0044] 여기서 B_I 및 $N_{0,I}$ 는 각각 IoT 기기와 사용자 단말(UE) 사이의 채널 대역폭과 잡음 전력 밀도를 나타낸다.

[0045] 유사하게 사용자 단말(UE)에서 MEC 서버로의 오프로딩 전송 속도(r_U)를 수학적 3에 따라 계산할 수 있다.

수학적 3

$$r_U = B_U \log_2 \left(1 + \frac{P_U \|h_U\|^2}{N_{0,U} B_U} \right)$$

[0046]

[0047] 여기서 B_U 및 $N_{0,U}$, P_U 및 h_U 는 각각 사용자 단말(UE)과 MEC 서버 사이의 채널 대역폭과 잡음 전력 밀도, 전송 전력 및 레일리 페이딩 블록 업링크 채널 이득을 나타낸다.

[0048] 그리고 각 태스크(I_k)에 대한 태스크 처리 시간을 계산할 수 있다.

[0049] 다수의 IoT 기기에서 발생된 태스크(I_k)는 사용자 단말(UE)에서 처리되거나 MEC 서버로 오프로딩되어 처리될 수 있으므로, 태스크 처리 시간 또한 사용자 단말(UE)에서 처리 시간 또한 사용자 단말에서 처리되는 로컬 처리 시간(T_l)과 MEC 서버로 오프로딩되어 처리되는 오프로딩 처리 시간(T_E)로 구분되어 계산될 수 있다.

[0050] IoT 기기에서 사용자 단말(UE)로의 태스크 전송 시간($T_{I,tr}$)은 태스크(I_k)의 데이터 크기(d_k)와 IoT 기기가 사용자 단말(UE)로 태스크(I_k)를 전송하는 태스크 전송 속도(r_I)에 따라 수학적 4로 계산될 수 있다.

수학적 4

$$T_{I,tr} = \frac{d_k}{r_I}$$

[0051]

[0052] 그리고 사용자 단말(UE)의 태스크 처리 시간($T_{U,pr}$)은 사용자 단말(UE)의 프로세서가 태스크를 처리하는데 요구되는 단말 태스크 처리 사이클 수($W_{k,U}$)와 프로세서의 주파수(f_U)에 따라 계산될 수 있다. 이때 본 실시예에서 사용자 단말(UE)은 동적 전압 및 주파수 스케일링(DVFS) 기술을 적용하여, 처리해야 할 태스크가 존재하면, 사용 가능한 주파수 $f_U \in F = \{f_{\min} = f_1, f_2, \dots, f_{\max}\}$ 중 가장 높은 최대 주파수(f_{\max})를 이용하고, 이외에는 에너지 소모를 저감하기 위해 가장 낮은 최소 주파수(f_{\min})를 이용하는 것으로 가정한다. 따라서 사용자 단말(UE)의 태스크 처리 시간($T_{U,pr}$)은 수학적 5와 같이 계산될 수 있다.

수학식 5

$$T_{U,pr} = \frac{W_{k,U}}{f_{max}}$$

[0053]

[0054] 로컬 처리 시간(T_U)은 IoT 기기에서 사용자 단말(UE)로의 태스크 전송 시간($T_{I,tr}$)과 사용자 단말(UE)의 태스크 처리 시간($T_{U,pr}$)의 합($T_U = T_{I,tr} + T_{U,pr}$)으로 수학식 6과 같이 계산될 수 있다.

수학식 6

$$T_U = \frac{d_k}{r_I} + \frac{W_{k,U}}{f_{max}}$$

[0055]

[0056] 그리고 오프로딩 처리 시간(T_E)은 IoT 기기에서 사용자 단말(UE)로의 태스크 전송 시간($T_{I,tr}$)과 사용자 단말(UE)에서 MEC 서버로의 오프로딩 전송 시간($T_{U,tr}$) 및 MEC 서버의 태스크 처리 시간($T_{E,pr}$)의 합($T_E = T_{I,tr} + T_{U,tr} + T_{E,pr}$)으로 계산될 수 있다.

[0057] 사용자 단말(UE)에서 MEC 서버로의 오프로딩 전송 시간($T_{U,tr}$)은 태스크(I_k)의 데이터 크기(d_k)와 오프로딩 전송 속도(r_U)에 따라 수학식 7로 계산될 수 있다.

수학식 7

$$T_{U,tr} = \frac{d_k}{r_U}$$

[0058]

[0059] 한편, MEC 서버에서 프로세서의 주파수가 f_E 이면, MEC 서버의 태스크 처리 시간($T_{E,pr}$)은 MEC 서버의 프로세싱 주파수(f_E)와 MEC 서버의 태스크 처리 사이클 수($W_{k,E}$)에 따라 수학식 8로 계산될 수 있다.

수학식 8

$$T_{E,pr} = \frac{W_{k,E}}{f_E}$$

[0060]

[0061] 따라서 오프로딩 처리 시간(T_E)은 수학식 9와 같이 계산될 수 있다.

수학식 9

$$T_E = \frac{d_k}{r_I} + \frac{d_k}{r_U} + \frac{W_{k,E}}{f_E}$$

[0062]

[0063] 그러므로 N개의 태스크 중 n개가 MEC 서버로 오프로딩되고 나머지는 사용자 단말(UE)에서 로컬 처리되는 경우의 총 서비스 시간($T(n)$)은 수학식 10과 같이 계산될 수 있다.

수학식 10

$$\mathcal{T}(n) = \sum_{k=1}^n T_E + \sum_{k=n+1}^N T_U$$

[0064]

[0065] 한편 태스크 오프로딩에 따른 에너지 소비량은 시간과 소비 전력의 곱으로 계산될 수 있으며, 전체 시스템의 총 에너지 소비량은 다수의 IoT 기기의 에너지 소비량(E_I)과 사용자 단말(UE)의 에너지 소비량(E_U)의 합으로 계산될 수 있다.

[0066] 또한 사용자 단말(UE)의 에너지 소비량을 계산하는 경우, 전력을 동적으로 이용할 수 있다는 것을 고려해야 한다. 즉 사용자 단말(UE)이 태스크 처리 시에 프로세서가 사용하는 프로세싱 전력(P_{pr})과 태스크를 전송할 때 사용하는 전송 전력(P_{tr}) 및 유휴 상태에서의 유휴 전력(P_{idle})을 구분하여 계산할 필요가 있다.

[0067] IoT 기기의 에너지 소비량(E_I)은 N개의 태스크(I_1, \dots, I_N)에 대한 사용자 단말(UE)로의 전송 에너지, N개의 태스크(I_1, \dots, I_N) 중 오프로딩되는 n개의 태스크(I_1, \dots, I_n)가 사용자 단말(UE)을 통해 MEC 서버로 오프로딩 처리되는 동안의 유휴 에너지 및 나머지 태스크(I_{n+1}, \dots, I_N)가 사용자 단말(UE)에서 로컬 처리되는 동안의 유휴 에너지의 합으로 수학식 11과 같이 계산될 수 있다.

수학식 11

$$\begin{aligned} E_I(n) = & \sum_{k=1}^N P_{tr} T_{I,tr} \\ & + \sum_{k=1}^n P_{idle} (T_{U,tr} + T_{E,pr}) \\ & + \sum_{k=n+1}^N P_{idle} T_{U,pr} \end{aligned}$$

[0068]

[0069] 여기서 P_{tr} 은 전송 전력, $T_{I,tr}$ 은 사용자 단말(UE)로의 태스크 전송 시간, P_{idle} 는 유휴 전력, $T_{U,tr}$ 은 사용자 단말(UE)로부터 MEC 서버로의 오프로딩 전송 시간($T_{U,tr}$), $T_{E,pr}$ 은 MEC 서버의 태스크 처리 시간, $T_{U,pr}$ 은 사용자 단말(UE)의 태스크 처리 시간을 나타낸다.

[0070] 한편, 사용자 단말(UE)의 에너지 소비량(E_U)은 N개의 태스크(I_1, \dots, I_N)가 전송되는 동안의 유휴 에너지, N개의 태스크(I_1, \dots, I_N) 중 오프로딩되는 n개의 태스크(I_1, \dots, I_n)를 MEC 서버로 전송하는 전송 에너지와 오프로딩된 n개의 태스크(I_1, \dots, I_n)가 MEC 서버에서 처리되는 동안의 유휴 에너지 및 사용자 단말(UE)이 나머지 태스크(I_{n+1}, \dots, I_N)를 로컬 처리하는 프로세싱 에너지의 합으로 수학식 12와 같이 계산될 수 있다.

수학식 12

$$E_U(n) = \sum_{k=1}^N P_{idle} T_{I,tr} + \sum_{k=1}^n (P_{tr} T_{U,tr} + P_{idle} T_{E,pr}) + \sum_{k=n+1}^N P_{pr} T_{U,pr}$$

[0071]

[0072]

여기서 P_{tr} 은 전송 전력, $T_{I,tr}$ 은 사용자 단말(UE)로의 태스크 전송 시간, P_{idle} 는 유휴 전력, $T_{U,tr}$ 은 사용자 단말(UE)로부터 MEC 서버로의 오프로딩 전송 시간($T_{U,tr}$), $T_{E,pr}$ 은 MEC 서버의 태스크 처리 시간, $T_{U,pr}$ 은 사용자 단말(UE)의 태스크 처리 시간을 나타낸다.

[0073]

따라서 MEC 서버 기반 오프로딩 시스템에서 N 개의 태스크(I_1, \dots, I_N) 중 n 개의 태스크가 오프로딩될 때, 총 에너지 소비($E(n)$)는 IoT 기기의 에너지 소비량(E_I)과 사용자 단말(UE)의 에너지 소비량(E_U)의 합($E(n) = E_I(n) + E_U(n)$)으로 계산될 수 있다.

[0074]

한편 이득 기여도 계산부(120)는 협력 게임 이론에 따라 N 개의 태스크(I_1, \dots, I_N)로부터 오프로딩될 태스크의 집합인 협동 연합(\mathbb{C})을 구성할 때, 발생하는 이득에 대한 각 태스크(I_k)의 기여도를 계산한다. 즉 N 개의 태스크(I_1, \dots, I_N)가 N 명의 플레이어로 협력하여 획득되는 추가로 획득된 전체 이득에서 개별 태스크(I_1, \dots, I_N)가 협동 연합(\mathbb{C})에 참여함 따른 획득된 이득 변화분을 나타내는 이득 기여도를 계산한다.

[0075]

N 개의 태스크(I_1, \dots, I_N)에서 태스크(I_i)가 협동 연합(\mathbb{C})에 참여함에 따른 이득 기여도(ϕ_i)는 협동 게임 이론의 샤플리 값(Shapley value)으로 수학식 13과 같이 계산될 수 있다.

수학식 13

$$\phi_i(v) = \sum_{\mathbb{C} \subseteq \mathbf{N} \setminus \{i\}} \frac{|\mathbb{C}|! (|\mathbf{N}| - |\mathbb{C}| - 1)!}{|\mathbf{N}|!} (v(\mathbb{C} \cup \{i\}) - v(\mathbb{C}))$$

[0076]

여기서 $\mathbb{C} \subseteq \mathbf{N} \setminus \{i\}$ 는 협동 연합(\mathbb{C})이 N 개의 태스크(I_1, \dots, I_N) 중 제 i 태스크(I_i)를 제외한 나머지 태스크의 조합으로 구성되는 모든 태스크 집합임을 의미하고, $v(\mathbb{C})$ 는 협동 연합(\mathbb{C})에 포함된 태스크가 전체 이득을 향상시키기 위해 협력함으로써 획득할 수 있는 이득을 계산하기 위한 특성 함수로서 협력 이득을 나타낸다.

[0077]

[0078]

수학식 13에 따르면, 이득 기여도(ϕ_i)는 태스크(I_i)가 협동 연합(\mathbb{C})에 참여한 경우($\mathbb{C} \cup \{i\}$)와 참여하지 않은 경우(\mathbb{C}) 사이의 협력 이득($v(\mathbb{C})$) 차이에 기반하여 계산된다.

[0079]

그리고 본 실시예의 오프로딩 시스템에서 이득은 IoT 기기와 사용자 단말(UE)의 에너지 소모를 저감시키는 것이므로, 협력 이득($v(\mathbb{C})$)은 협동 연합(\mathbb{C})에 포함되는 태스크의 구성에 따른 협력 에너지 소비량($E_{\mathbb{C}}$)에 기반하여 수학식 14와 같이 계산될 수 있다.

수학식 14

$$v(\mathbb{C}) = \gamma \sum_{i \in \mathbb{C}} f_o(i) E_{\mathbb{C}}$$

[0080]

[0081]

여기서 $f_o(i)$ 는 협동 연합(\mathbb{C})에 포함되는 각 태스크(I_i)의 발생 빈도를 나타내고, γ 는 협력에 대한 이익 전환율(payload conversion rate)을 나타내는 밸런스 파라미터(balancing parameter)로서 미리 지정된 상수 값으로 설정될 수 있다. 그리고 $E_{\mathbb{C}}$ 는 N 개의 태스크(I_1, \dots, I_N)에서 협동 연합(\mathbb{C})이 오프로딩되어 MEC 서버에서 처리될 때의 총 에너지 소비량을 나타낸다. 협동 연합(\mathbb{C})에 포함되는 태스크의 개수가 n 개일 때, 협력 에너지 소비량($E_{\mathbb{C}}$)은 MEC 서버에서 처리되는 각 태스크에 대한 처리 에너지의 합($E_{\mathbb{C}} = \sum_{k=1}^n P_{pr} T_{E,pr}$)으로 계산될 수 있다.

[0082]

태스크 정렬부(130)는 이득 기여도 계산부(120)에 의해 N 개의 태스크(I_1, \dots, I_N) 각각에 대한 이득 기여도(ϕ_i)가 계산되면, 계산된 이득 기여도(ϕ_i)에 따라 N 개의 태스크(I_1, \dots, I_N)를 우선 순위 정렬한다. 태스크 정렬부(130)는 N 개의 태스크(I_1, \dots, I_N)에서 이득 기여도(ϕ_i)가 가장 큰 태스크부터 내림차순 순서로 순차 정렬한다.

[0083]

한편 시간 감소량 계산부(140)는 이득 기여도 계산부(120)에 의해 N 개의 태스크(I_1, \dots, I_N) 각각에 대한 이득 기여도(ϕ_k)를 기반으로 각 태스크(I_k)를 오프로딩 하여 저장되는 감소 시간(T_R)을 계산한다. 이득 기여도(ϕ_k)가 각 태스크를 처리함에 따른 에너지 이득을 나타내므로, 각 태스크(I_k)의 처리에 대한 시간 감소량(T_R)은 수학식 15와 같이 계산될 수 있다.

수학식 15

$$T_R(I_k) = \frac{\varphi_k(v)}{P_{pr}}$$

[0084]

[0085]

시간 감소량이 계산되면, 협력에 의해 감소된 총 서비스 시간($T_{co}(n)$)은 수학식 16과 같이 계산될 수 있다.

수학식 16

$$T_{co}(n) = \sum_{k=1}^n (T_E - T_R) + \sum_{k=n+1}^N T_U$$

[0086]

[0087]

오프로딩 태스크 결정부(160)는 수학식 16에 따라 계산된 시간 감소량(T_R)을 이용하여 에너지 소비가 최소화되도록 오프로딩될 태스크의 개수(n^*)를 계산하고, 정렬된 N 개의 태스크(I_1, \dots, I_N)에서 정렬 순서에 따라 계산된 태스크 개수(n^*)의 태스크가 오프로딩되고 나머지 태스크를 사용자 단말(UE)이 로컬로 처리되도록 오프로딩을 태스크를 결정한다.

[0088]

수학식 16에 따라 시간 감소량(T_R)이 계산되면, 협력에 의한 시간 감소량(T_R)을 추가로 고려한 IoT 기기의 에너지 소비량(E_I)과 사용자 단말(UE)의 에너지 소비량(E_U)은 각각 수학식 11 및 12로부터 수학식 17 및 18과 같이 수정되어 계산될 수 있다.

수학식 17

$$E_{I_co}(n) = \sum_{k=1}^N P_{tr} T_{I,tr} + \sum_{k=1}^n P_{idle} (T_{U,tr} + T_{E,pr} - T_R) + \sum_{k=n+1}^N P_{idle} T_{U,pr}$$

수학식 18

$$E_{U_co}(n) = \sum_{k=1}^N P_{idle} T_{I,tr} + \sum_{k=1}^n (P_{tr} T_{U,tr} + P_{idle} (T_{E,pr} - T_R)) + \sum_{k=n+1}^N P_{pr} T_{U,pr}$$

수학식 17 및 18과 같이 협력에 따른 IoT 기기의 에너지 소비량(E_{I_co})과 사용자 단말(UE)의 에너지 소비량(E_{U_co})이 계산되면, 협력 시의 총 에너지 소비 프로파일($\mathcal{E}(n)$)은 IoT 기기와 사용자 단말(UE)의 에너지 소비량(E_{I_co} , E_{U_co})의 합으로 수학식 19로 나타낼 수 있다.

수학식 19

$$\mathcal{E}(n) = E_{I_co}(n) + E_{U_co}(n)$$

오프로딩의 목적은 총 에너지 소비량을 저감하기 위한 것이므로, 총 에너지 소비 프로파일($\mathcal{E}(n)$)이 최소가 되도록 하는 n 을 탐색하는 목적함수는 서비스의 QoS를 만족시키기 위한 제약 조건(C1 ~ C3)을 고려하여 수학식 20으로 정리될 수 있다.

수학식 20

$$\begin{aligned} \min_n \quad & \mathcal{E}(n) \\ \text{s.t.} \quad & C1 : \mathcal{T}_{co}(n) \leq \mathcal{T}_{th} \\ & C2 : 0 \leq n \leq N \\ & C3 : 0 \leq \gamma \leq 0.5 \end{aligned}$$

수학식 20에서 제1 제약 조건(C1)은 총 서비스 시간이 서비스 제한 시간(\mathcal{T}_{th}) 이하이어야 한다는 조건을 나타내고, 제1 제약 조건(C1)은 오프로딩될 태스크의 개수(n)는 전체 태스크의 개수(N) 이하이어야 한다는 조건을 나타내며, 제3 제약 조건(C3)은 각 태스크(I_i)의 발생 빈도($f_o(i)$)에 따른 협력에 대한 이익 전환율을 나타내는 벨런스 파라미터(γ)가 0 ~ 0.5 사이의 범위를 가져야함을 나타낸다.

사용자 단말(UE)이 오프로딩하는 태스크의 개수(n)가 많을수록 에너지 소비는 저감되지만, 수학식 16의 감소된 총 서비스 시간($\mathcal{T}_{co}(n)$)이 서비스 제한 시간(\mathcal{T}_{th}) 이하이어야 하므로, 제1 제약 조건(C1)은 수학식 21로 표현될 수 있다.

수학식 21

$$\sum_{k=1}^n (T_E(I_k) - T_R(I_k)) + \sum_{k=n+1}^N T_U(I_k) \leq T_{th}$$

그리고 수학식 21로부터 수학식 20을 만족하는 오프로딩될 태스크의 개수(n)는 수학식 22로 표현될 수 있다.

수학식 22

$$n(T_E(I_k) - T_R(I_k)) + (N - n)T_U(I_k) \leq T_{th},$$

$$n^* = \left\lfloor \frac{T_{th} - NT_U(I_k)}{T_E(I_k) - T_U(I_k) - T_R(I_k)} \right\rfloor.$$

여기서 $\lfloor \cdot \rfloor$ 는 바닥 함수(또는 마루 함수)를 나타낸다.

이에 오프로딩 태스크 결정부(160)는 수학식 22에 따라 오프로딩될 태스크 개수(n^*)가 계산되면, 정렬된 N개의 태스크(I_1, \dots, I_N)의 정렬 순서에 따라 계산된 태스크 개수(n^*)의 태스크를 오프로딩하고 나머지 태스크를 사용자 단말(UE)이 로컬로 처리하도록 한다.

도 3은 본 발명의 일 실시예에 다른 태스크 오프로딩 방법을 나타낸다.

도 1 및 도 2를 참조하여, 도 3의 태스크 오프로딩 방법을 설명하면, 우선 오프로딩할 태스크를 선택하기 위해 필요한 정보를 수집한다(S10). 여기서 수집되는 정보에는 사용자 단말(UE)에 대응하여 클러스터링된 다수의 IOT 기기에서 전송되는 태스크 정보와 통신 정보 및 사용자 단말(UE)과 MEC 서버의 연산 능력 정보가 포함될 수 있다.

그리고 협력 게임 이론에 따라 N개의 태스크(I_1, \dots, I_N)로부터 오프로딩될 태스크의 집합인 협동 연합(\mathbb{C})을 구성할 때, 발생하는 이득에 대한 각 태스크(I_k)의 이득 기여도(ϕ_i)를 수집된 정보를 이용하여 계산한다(S20). 여기서 태스크(I_k)의 이득 기여도(ϕ_i)는 샤플리 값으로 수학식 13과 같이 계산될 수 있으며, 이때 이득 기여도(ϕ_i)를 계산하기 위한 특성 함수($v(\mathbb{C})$)는 협동 연합(\mathbb{C})에 포함된 태스크가 전체 이득을 향상시키기 위해 협력함으로써 획득할 수 있는 이득을 계산하기 위한 협력 이득으로서 수학식 14에 따라 계산될 수 있다.

N개의 태스크(I_1, \dots, I_N) 각각에 대한 이득 기여도(ϕ_i)가 계산되면, 계산된 이득 기여도(ϕ_i)에 따라 N개의 태스크(I_1, \dots, I_N)를 우선 순위 정렬한다(S30). 즉 N개의 태스크(I_1, \dots, I_N)에서 이득 기여도(ϕ_i)가 가장 큰 태스크부터 내림차순 순서로 순차 정렬한다. 여기서 정렬된 순서는 해당 태스크가 오프로딩될 오프로딩 우선 순위이다.

이와 함께 N개의 태스크(I_1, \dots, I_N) 각각에 대한 이득 기여도(ϕ_k)를 기반으로 각 태스크(I_k)를 오프로딩 하여 저장되는 감소 시간(T_R)을 계산한다(S40). 여기서 감소 시간(T_R)은 이득 기여도(ϕ_k)와 태스크(I_k)를 처리하기 위해 요구되는 프로세싱 전력(P_{pr})으로부터 수학식 15와 같이 계산할 수 있다.

감소 시간(T_R)이 계산되면, 서비스 제한 시간(T_{th})을 만족해야 하는 조건에서 감소 시간(T_R)을 고려한 다수의 IoT 기기와 사용자 단말(UE)의 에너지 소비량의 합을 나타내는 협력 시의 총 에너지 소비 프로파일($\mathcal{E}(n)$)이 최소가 되도록 오프로딩 개수(n^*)를 계산한다. 여기서 오프로딩 개수(n^*)는 수학식 22에 따라 계산될 수 있다.

본 발명에 따른 방법은 컴퓨터에서 실행시키기 위한 매체에 저장된 컴퓨터 프로그램으로 구현될 수 있다. 여기

서 컴퓨터 판독가능 매체는 컴퓨터에 의해 액세스 될 수 있는 임의의 가용 매체일 수 있고, 또한 컴퓨터 저장 매체를 모두 포함할 수 있다. 컴퓨터 저장 매체는 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 기타 데이터와 같은 정보의 저장을 위한 임의의 방법 또는 기술로 구현된 휘발성 및 비휘발성, 분리형 및 비분리형 매체를 모두 포함하며, ROM(판독 전용 메모리), RAM(랜덤 액세스 메모리), CD(컴팩트 디스크)-ROM, DVD(디지털 비디오 디스크)-ROM, 자기 테이프, 플로피 디스크, 광데이터 저장장치 등을 포함할 수 있다.

[0109] 본 발명은 도면에 도시된 실시예를 참고로 설명되었으나 이는 예시적인 것에 불과하며, 본 기술 분야의 통상의 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 타 실시예가 가능하다는 점을 이해할 것이다.

[0110] 따라서, 본 발명의 진정한 기술적 보호 범위는 첨부된 청구범위의 기술적 사상에 의해 정해져야 할 것이다.

부호의 설명

[0111] 110: 정보 수집부 120: 이득 기여도 계산부

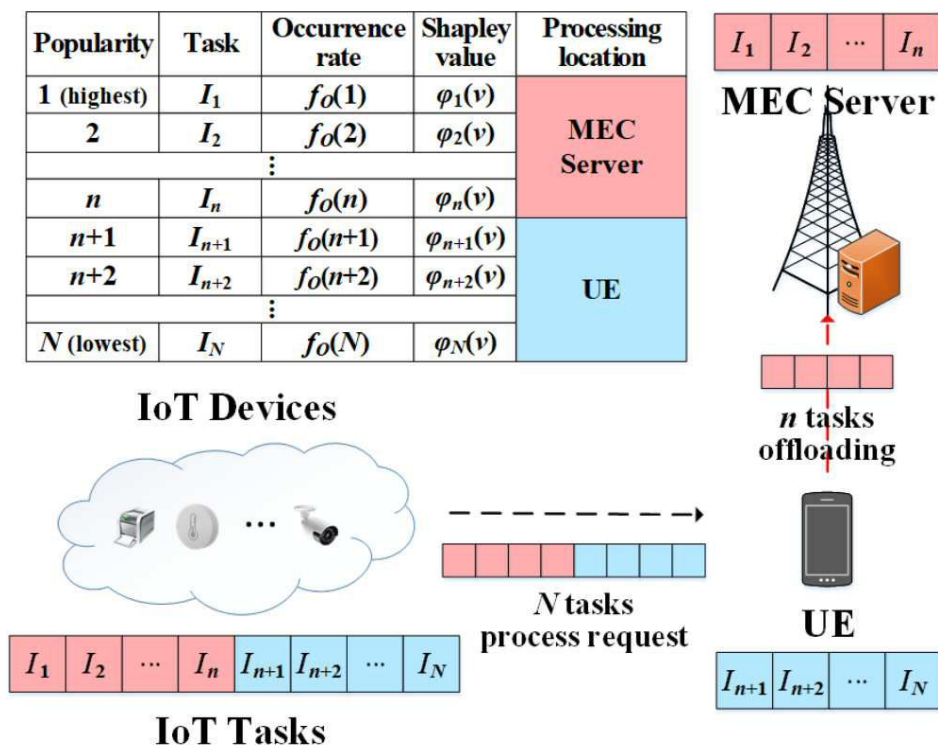
130: 태스크 정렬부

140: 시간 감소량 계산부

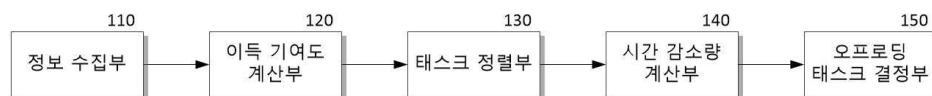
150: 오프로딩 태스크 결정부

도면

도면1



도면2



도면3

