(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2023년01월26일
(11) 등록번호 10-2492716
(24) 등록일자 2023년01월20일

(51) 국제특허분류(Int. Cl.)
G06F 9/48 (2018.01) *G06F 9/455* (2018.01)
H04L 1/00 (2006.01)
(52) CPC특허분류
G06F 9/4881 (2013.01)
G06F 9/45558 (2013.01)
(21) 출원번호 10-2021-0077476
(22) 출원일자 2021년06월15일
심사청구일자 2021년06월15일
(65) 공개번호 10-2022-0168001
(43) 공개일자 2022년12월22일
(56) 선행기술조사문헌
KR1020210064031 A
KR1020200062272 A
KR102177432 B1

(73) 특허권자
연세대학교 산학협력단
서울특별시 서대문구 연세로 50 (신촌동, 연세대학교)
(72) 발명자
정종문
서울특별시 용산구 이촌로 181, 104동 101호(이촌동, 한강대우아파트)
윤주식
서울특별시 서대문구 신촌로7길 49-6, 202호(창천동, 청송빌)
(뒷면에 계속)
(74) 대리인
민영준

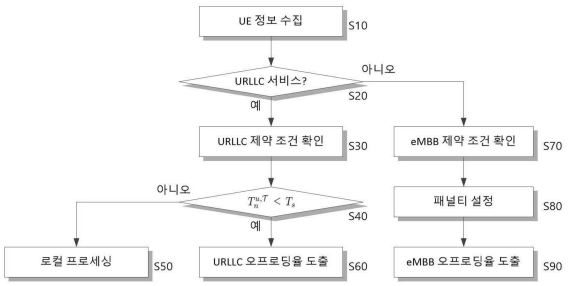
전체 청구항 수 : 총 20 항 심사관 : 지정훈

(54) 발명의 명칭 **다중 라디오 액세스 기술 기반 IIoT 서비스를 위한 동적 태스크 오프로딩을 수행하는 사용자 단말 및 이의 오프로딩 방법**

(57) 요 약

본 발명은 요구사항이 서로 다른 다수의 서비스 각각의 특성과 요구 사항 및 사용자 단말의 에너지 소모를 고려하여 오프로딩 비율을 동적으로 결정함으로써 각 서비스에서 요구되는 QoS를 만족시키면서 에너지 소비를 저감하고 성능을 최적화할 수 있는 사용자 단말 및 이의 오프로딩 방법을 제공한다.

대 표 도



(52) CPC특허분류
G06F 9/4825 (2013.01)
H04L 1/0018 (2013.01)
H04L 67/61 (2022.05)
H04L 67/62 (2022.05)
G06F 2009/45595 (2019.08)

(72) 발명자
고윤영
서울특별시 서대문구 신촌로7길 49-15(창천동)

유원석
경기도 부천시 양지로92번길 33, 안팰리스 2차 306호(범박동)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711126081
과제번호	2018-0-01799-004
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	정보통신방송혁신인재양성(R&D)
연구과제명	블록체인 비즈니스 서비스 기술 개발 및 인력양성
기 여 율	1/1
과제수행기관명	중앙대학교산학협력단
연구기간	2021.01.01 ~ 2021.12.31

명 세 서

청구범위

청구항 1

MEC(Multi-access Edge Computing) 환경의 다중 RAT(Radio Access Technology) 네트워크에 포함되는 사용자 단말로서,

다중 RAT를 통해 MEC 서버와 통신을 수행하는 통신부;

상기 사용자 단말이 이용하는 서비스에 따라 처리되어야 하는 태스크 패킷이 임시 저장되는 로컬 큐;

상기 로컬 큐에 저장된 태스크 패킷 각각에 연산 자원인 로컬 CPU 사이클을 할당하여 처리하는 로컬 프로세서; 및

상기 서비스에 따라 획득된 태스크 패킷 중 상기 로컬 큐로 전달되어 상기 로컬 프로세서에서 처리되는 로컬 프로세싱율과 상기 통신부를 통해 상기 MEC 서버로 오프로딩되어 MEC 서버에서 처리되는 오프로딩율을 결정하는 오프로딩 스케쥴러를 포함하되,

상기 오프로딩 스케쥴러는 사용자 단말이 이용하는 서비스가 URLLC 서비스인 경우, 비트 단위로 URLLC 로컬 프로세싱율($o_{n,0}^{u}$)에 따른 에너지 소비($E_n^{u,L}$)와 URLLC 오프로딩율($1 - o_{n,0}^{u}$)에 따라 오프로딩되는 태스크 패킷인 URLLC 패킷의 전송 지연 시간($T_n^{u,\mathcal{T}}$)동안 단일 타임 슬롯(T_s) 단위의 전송 전력($p_{n,m}^{u}$)의 합으로 계산되는 정규화된 URLLC 에너지 소비량(E_n^{u})이 최소가 되는 URLLC 오프로딩율($1 - o_{n,0}^{u}$)을 유전 알고리즘을 이용하여 획득하는 사용자 단말.

청구항 2

제1항에 있어서, 상기 오프로딩 스케쥴러는

상기 URLLC 서비스의 태스크 각각 대한 태스크 패킷은 단일 URLLC 패킷으로 구성되고 로컬 프로세싱되거나 하나의 RAT를 통해 오프로딩되어, 상기 URLLC 패킷의 전송 지연($T_n^{u,\mathcal{T}}$)이 1 타임 슬롯(T_s) 이상이면, URLLC 패킷을 사용자 단말이 로컬 프로세싱하도록 URLLC 오프로딩율($1 - o_{n,0}^{u}$)을 0으로 획득하는 사용자 단말.

청구항 3

제2항에 있어서, 상기 오프로딩 스케쥴러는

상기 URLLC 패킷의 전송 지연($T_n^{u,\mathcal{T}}$)이 1 타임 슬롯(T_s) 미만이면,

수학식

$$F_n^u = \frac{o_{n,0}^u E_n^{u,\mathcal{L}} + (1-o_{n,0}^u) p_{n,m}^u T_n^{u,\mathcal{T}} T_s}{b_n^u}$$

(여기서 F_n^{u}은 URLLC 패킷에 대한 정규화된 에너지 소비량($F_n^{u} = E_n^{u}$), b_n^{u} 는 URLLC 패킷 크기)

이 최소가 되도록 하는 URLLC 오프로딩율($1 - o_{n,0}^{u}$)을 획득하는 사용자 단말.

청구항 4

제3항에 있어서, 상기 오프로딩 스케쥴러는

상기 유전 알고리즘을 이용하여 URLLC 오프로딩율($1 - o_{n,0}^{u}$)을 획득함에 있어,

URLLC 패킷의 전체 손실 확률(ε_n^u)이 URLLC 서비스에 의해 기지정된 최대 손실 확률($\varepsilon_n^{u,max}$)이어야 하고, 단일 RAT를 통해 전송되는 URLLC 패킷의 전송 전력($p_{n,m}^u$)의 합이 사용자 단말에 기지정된 최대 전송 전력(p_n^{max}) 이하이어야 하며, 사용자 단말의 로컬 CPU 사이클(f_n)이 기지정된 최소 CPU 사이클(f_n^{min})과 최대 CPU 사이클(f_n^{max}) 사이의 값을 가지고, 로컬 프로세싱을 포함한 다수의 RAT를 통한 URLLC 오프로딩율의 총합은 1이다라는 제약 조건하에서 URLLC 패킷에 대한 정규화된 에너지 소비량(E_n^u)이 최소가 되는 URLLC 오프로딩율($1 - o_{n,0}^{u}$)을 획득하는 사용자 단말.

청구항 5

제4항에 있어서, 상기 오프로딩 스케쥴러는

상기 URLLC 패킷에 대한 정규화된 에너지 소비량(E_n^u)이 최소가 되는 URLLC 오프로딩율($1 - o_{n,0}^{u}$)과 함께 사용자 단말의 로컬 프로세싱을 위해 할당되어야 하는 로컬 CPU 사이클(f_n) 및 다수의 RAT 중 하나의 RAT를 통해 URLLC 패킷을 오프로딩하기 위한 URLLC 전송 전력($p_{n,m}^u$)을 유전 알고리즘을 이용하여 획득하는 사용자 단말.

청구항 6

제1항에 있어서, 상기 오프로딩 스케쥴러는

상기 사용자 단말이 이용하는 서비스가 eMBB 서비스인 경우, 비트 단위로 사용자 단말이 eMBB 태스크가 분할된 다수의 eMBB 패킷 중 직접 eMBB 패킷을 처리하는 eMBB 로컬 프로세싱율($o_{n,0}^e$)에 따른 eMBB 로컬 에너지 소비량($E_n^{e,L}$)과 다수의 RAT 각각에 대한 eMBB 오프로딩율($o_{n,m}^e$)에 따라 eMBB 패킷을 MEC 서버로 오프로딩하기 위한 전송 에너지 소비량($E_n^{e,\mathcal{T}}$)의 합으로 계산되는 정규화된 eMBB 에너지 소비량(E_n^e)이 최소가 되는 eMBB 오프로딩율($o_{n,0}^e$, $o_{n,m}^e$)을 유전 알고리즘을 이용하여 획득하는 사용자 단말.

청구항 7

제6항에 있어서, 상기 오프로딩 스케쥴러는

사용자 단말의 로컬 프로세싱에 따른 로컬 프로세싱 시간($T_{n,P}^{e,M}$)과 다수의 RAT 각각 통한 오프로딩 전송 지연 시간($T_{n,i}^{e,\mathcal{T}}$) 사이의 시간차 및 다수의 RAT를 통한 오프로딩 시의 각 RAT 사이의 전송 지연 시간차($T_{n,i}^{e,\mathcal{T}} - T_{n,j}^{e,\mathcal{T}}$)따라 계산되는 패널티($v_n$)를 상기 정규화된 eMBB 에너지 소비량($E_n^e$)에 가산한 결과가 최소가 되도록 eMBB 오프로딩율($o_{n,0}^u$, $o_{n,m}^u$)을 유전 알고리즘을 이용하여 획득하는 사용자 단말.

청구항 8

제7항에 있어서, 상기 오프로딩 스케쥴러는

상기 MEC 서버가 다수의 사용자 단말 각각에서 오프로딩되는 태스크 패킷 중 URLLC 패킷과 eMBB 패킷을 서로 구분하여 저장하도록 독립적으로 구성되는 제1 태스크 큐 및 제2 태스크 큐를 구비하는 것으로 판단하는 사용자 단말.

청구항 9

제8항에 있어서, 상기 오프로딩 스케쥴러는

상기 패널티(v_n)를 수학식

$$\vartheta_n = \sum_{i,j \in M} \sqrt{(T_{n,\mathcal{P}}^{e,\mathcal{L}} - T_{n,i}^{e,\mathcal{T}})^2} + \sqrt{(T_{n,i}^{e,\mathcal{T}} - T_{n,j}^{e,\mathcal{T}})^2}$$

(여기서 $T_{n,P}^{e,L}$ 는 사용자 단말이 eMBB 패킷을 프로세싱하는데 소요되는 로컬 eMBB 프로세싱 시간, $T_{n,i}^{e,\mathcal{T}}$와 $T_{n,j}^{e,\mathcal{T}}$ 는 각각 다수의 RAT 중 i번째 및 j번째 RAT를 통한 오프로딩 전송 지연 시간)

에 따라 계산하고,

수학식

$$F_n^e = \frac{k_e (f_n)^2 o_{n,0}^e c_n^e + \sum_{m \in M} \frac{p_{n,m}^e b_n^e o_{n,m}^e}{r_{n,m}^e}}{b_n^e} + \Omega \vartheta_n$$

(여기서 k_e 는 CPU의 처리 능력에 따라 달라지는 에너지 계수, f_n 은 사용자 단말의 로컬 CPU 사이클, c_n^e 는 MEC 서버가 오프로딩되어 제2 태스크 큐에 저장된 eMBB 패킷을 프로세싱하는데 요구되는 CPU 사이클, M은 RAT 개수, p_n^e 는 $p_{n,m}^e$은 m번째 RAT를 통해 오프로딩 시 사용자 단말의 eMBB 패킷 전송 전력, b_n^e는 eMBB 패킷 크기, $o_{n,m}^e$ 는 m번째 RAT를 통한 사용자 단말의 eMMB 오프로딩율, $r_{n,m}^e$ 는 eMBB 데이터율($r_{n,m}^e$), Ω는 패널티(v_n)의 반영 수준을 조절하기 위한 가중치이다.)

이 최소가 되도록 하는 eMBB 오프로딩율($o_{n,0}^e$, $o_{n,m}^e$)을 획득하는 사용자 단말.

청구항 10

제9항에 있어서, 상기 오프로딩 스케쥴러는

상기 유전 알고리즘을 이용하여 eMBB 오프로딩율($o_{n,0}^e$, $o_{n,m}^e$)을 획득함에 있어, eMBB 태스크의 총 지연 시간(T_n^e)이 기지정된 eMBB 최대 허용 대기 시간($T_n^{e,max}$) 이하이어야 하고, 다수의 RAT를 통해 전송되는 eMBB 패킷의 전송 전력($p_{n,m}^u$)의 합이 사용자 단말에 기지정된 최대 전송 전력(p_n^{max}) 이하이어야 하며, 사용자 단말의 로컬 CPU 사이클(f_n)이 기지정된 최소 CPU 사이클(f_n^{min})과 최대 CPU 사이클(f_n^{max}) 사이의 값을 가지고, 로컬 프로세싱을 포함한 다수의 RAT를 통한 오프로딩율의 총합은 1이다라는 제약 조건하에서 eMBB 패킷에 대한 수학식(F_n^e)이 최소가 되는 오프로딩율($o_{n,0}^e$, $o_{n,m}^e$)을 획득하는 사용자 단말.

청구항 11

제10항에 있어서, 상기 오프로딩 스케쥴러는

상기 eMBB 패킷에 대한 수학식(F_n^e)이 최소가 되는 오프로딩율($o_{n,0}^e$, $o_{n,m}^e$)과 함께 사용자 단말의 로컬 프로세싱을 위해 할당되어야 하는 로컬 CPU 사이클(f_n) 및 다수의 RAT를 통해 eMBB 패킷을 오프로딩하기 위한 eMBB 전송 전력($p_{n,m}^e$)을 유전 알고리즘을 이용하여 획득하는 사용자 단말.

청구항 12

MEC(Multi-access Edge Computing) 환경의 다중 RAT(Radio Access Technology) 네트워크에 포함되는 사용자 단말의 오프로딩 방법에 있어서,

다중 RAT를 통해 MEC 서버와 통신을 수행하여 MEC 서버와 다수의 RAT의 채널 상태 및 사용자 단말의 정보를 수

집하는 단계;

상기 사용자 단말이 이용하는 서비스를 판별하는 단계;

사용자 단말이 이용하는 서비스가 URLLC 서비스이면, 상기 URLLC 서비스의 태스크 각각에 대응하는 단일 URLLC 패킷을 다수의 RAT 중 하나의 RAT를 통해 상기 MEC 서버로 오프로딩 시에 발생하는 전송 지연($T_n^{u,\mathcal{T}}$)이 1 타임 슬롯(T_s) 이상인지 판별하는 단계;

상기 URLLC 패킷의 전송 지연($T_n^{u,\mathcal{T}}$)이 1 타임 슬롯(T_s) 이상이면, URLLC 패킷을 사용자 단말이 로컬 프로세싱하도록 URLLC 오프로딩율($1 - o_{n,0}^u$)을 0으로 획득하는 단계; 및

상기 URLLC 패킷의 전송 지연($T_n^{u,\mathcal{T}}$)이 1 타임 슬롯(T_s) 미만이면, 유전 알고리즘을 이용하여 비트 단위로 URLLC 로컬 프로세싱율($o_{n,0}^u$)에 따른 에너지 소비($E_n^{u,L}$)와 URLLC 오프로딩율($1 - o_{n,0}^u$)에 따라 오프로딩되는 태스크 패킷인 URLLC 패킷의 전송 지연 시간($T_n^{u,\mathcal{T}}$)동안 단일 타임 슬롯(T_s) 단위의 전송 전력($p_{n,m}^u$)의 합으로 계산되는 정규화된 URLLC 에너지 소비량(E_n^u)이 최소가 되는 URLLC 오프로딩율($1 - o_{n,0}^u$)을 획득하는 단계를 포함하는 사용자 단말의 오프로딩 방법.

청구항 13

제12항에 있어서, 상기 URLLC 오프로딩율($1 - o_{n,0}^u$)을 획득하는 단계는

수학식

$$F_n^u = \frac{o_{n,0}^u E_n^{u,\mathcal{L}} + (1-o_{n,0}^u) p_{n,m}^u T_n^{u,\mathcal{T}} T_s}{b_n^u}$$

(여기서 F_n^u은 URLLC 패킷에 대한 정규화된 에너지 소비량($F_n^u = E_n^u$), b_n^u 는 URLLC 패킷 크기)

이 최소가 되도록 하는 URLLC 오프로딩율($1 - o_{n,0}^u$)을 획득하는 사용자 단말의 오프로딩 방법.

청구항 14

제13항에 있어서, 상기 URLLC 오프로딩율($1 - o_{n,0}^u$)을 획득하는 단계는

URLLC 패킷의 전체 손실 확률(ε_n^u)이 URLLC 서비스에 의해 기지정된 최대 손실 확률($\varepsilon_n^{u,max}$)이어야 하고, 단일 RAT를 통해 전송되는 URLLC 패킷의 전송 전력($p_{n,m}^u$)의 합이 사용자 단말에 기지정된 최대 전송 전력(p_n^{max}) 이하이어야 하며, 사용자 단말의 로컬 CPU 사이클(f_n)이 기지정된 최소 CPU 사이클(f_n^{min})과 최대 CPU 사이클(f_n^{max}) 사이의 값을 가지고, 로컬 프로세싱을 포함한 다수의 RAT를 통한 URLLC 오프로딩율의 총합은 1이다라는 제약 조건하에서 URLLC 패킷에 대한 정규화된 에너지 소비량(E_n^u)이 최소가 되는 URLLC 오프로딩율($1 - o_{n,0}^u$)을 획득하는 사용자 단말의 오프로딩 방법.

청구항 15

제13항에 있어서, 상기 사용자 단말의 오프로딩 방법은

상기 사용자 단말이 이용하는 서비스가 eMBB 서비스인 경우, 유전 알고리즘을 이용하여, 비트 단위로 사용자 단말이 eMBB 태스크가 분할된 다수의 eMBB 패킷 중 직접 eMBB 패킷을 처리하는 eMBB 로컬 프로세싱율($o_{n,0}^e$)에 따

른 eMBB 로컬 에너지 소비량($E_n^{e,L}$)과 다수의 RAT 각각에 대한 eMBB 오프로딩율($o_{n,m}^{e}$)에 따라 eMBB 패킷을 MEC 서버로 오프로딩하기 위한 전송 에너지 소비량($E_n^{e,\mathcal{T}}$)의 합으로 계산되는 정규화된 eMBB 에너지 소비량(E_n^{e})이 최소가 되는 eMBB 오프로딩율($o_{n,0}^{e}$, $o_{n,m}^{e}$)을 획득하는 단계를 더 포함하는 사용자 단말의 오프로딩 방법.

청구항 16

제15항에 있어서, 상기 eMBB 오프로딩율($o_{n,0}^{e}$, $o_{n,m}^{e}$)을 획득하는 단계는

사용자 단말의 로컬 프로세싱에 따른 로컬 프로세싱 시간($T_{n,P}^{e,M}$)과 다수의 RAT 각각 통한 오프로딩 전송 지연 시간($T_{n,i}^{e,\mathcal{T}}$) 사이의 시간차 및 다수의 RAT를 통한 오프로딩 시의 각 RAT 사이의 전송 지연 시간차($T_{n,i}^{e,\mathcal{T}} - T_{n,j}^{e,\mathcal{T}}$)에 따라 계산되는 패널티($v_n$)를 상기 정규화된 eMBB 에너지 소비량($E_n^{e}$)에 가산한 결과가 최소가 되도록 eMBB 오프로딩율($o_{n,0}^{u}$, $o_{n,m}^{u}$)을 유전 알고리즘을 이용하여 획득하는 사용자 단말의 오프로딩 방법.

청구항 17

제16항에 있어서, 상기 eMBB 오프로딩율($o_{n,0}^{e}$, $o_{n,m}^{e}$)을 획득하는 단계는

상기 MEC 서버가 다수의 사용자 단말 각각에서 오프로딩되는 태스크 패킷 중 URLLC 패킷과 eMBB 패킷을 서로 구분하여 저장하도록 독립적으로 구성되는 제1 태스크 큐 및 제2 태스크 큐를 구비하는 것으로 판단하여 상기 eMBB 오프로딩율($o_{n,0}^{e}$, $o_{n,m}^{e}$)을 획득하는 사용자 단말의 오프로딩 방법.

청구항 18

제17항에 있어서, 상기 eMBB 오프로딩율($o_{n,0}^{e}$, $o_{n,m}^{e}$)을 획득하는 단계는

상기 패널티(v_n)를 수학식

$$\vartheta_n = \sum_{i,j \in M} \sqrt{(T_{n,\mathcal{P}}^{e,\mathcal{L}} - T_{n,i}^{e,\mathcal{T}})^2} + \sqrt{(T_{n,i}^{e,\mathcal{T}} - T_{n,j}^{e,\mathcal{T}})^2}$$

(여기서 $T_{n,P}^{e,L}$ 는 사용자 단말이 eMBB 패킷을 프로세싱하는데 소요되는 로컬 eMBB 프로세싱 시간, $T_{n,i}^{e,\mathcal{T}}$와 $T_{n,j}^{e,\mathcal{T}}$ 는 각각 다수의 RAT 중 i번째 및 j번째 RAT를 통한 오프로딩 전송 지연 시간)

에 따라 계산하고,

수학식

$$F_n^e = \frac{k_e (f_n)^2 o_{n,0}^e c_n^e + \sum_{m \in M} \frac{p_{n,m}^e b_n^e o_{n,m}^e}{r_{n,m}^e}}{b_n^e} + \Omega \vartheta_n$$

(여기서 k_e 는 CPU의 처리 능력에 따라 달라지는 에너지 계수, f_n 은 사용자 단말의 로컬 CPU 사이클, c_n^{e} 는 MEC 서버가 오프로딩되어 제2 태스크 큐에 저장된 eMBB 패킷을 프로세싱하는데 요구되는 CPU 사이클, M은 RAT 개수, p_n^{e} 는 $p_{n,m}^{e}$은 m번째 RAT를 통해 오프로딩 시 사용자 단말의 eMBB 패킷 전송 전력, b_n^{e}는 eMBB 패킷 크기, $o_{n,m}^{e}$ 는 m번째 RAT를 통한 사용자 단말의 eMMB 오프로딩율, $r_{n,m}^{e}$ 는 eMBB 데이터율($r_{n,m}^{e}$), Ω는 패널티(v_n)의 반영 수준을 조절하기 위한 가중치이다.)

이 최소가 되도록 하는 eMBB 오프로딩율($o_{n,0}^{e}$, $o_{n,m}^{e}$)을 획득하는 사용자 단말의 오프로딩 방법.

청구항 19

제18항에 있어서, 상기 eMBB 오프로딩율($o_{n,0}^{e}$, $o_{n,m}^{e}$)을 획득하는 단계는

상기 유전 알고리즘을 이용하여 eMBB 오프로딩율($o_{n,0}^{e}$, $o_{n,m}^{e}$)을 획득함에 있어, eMBB 태스크의 총 지연 시간(T_n^{e})이 기지정된 eMBB 최대 허용 대기 시간($T_n^{e,max}$) 이하이어야 하고, 다수의 RAT를 통해 전송되는 eMBB 패킷의 전송 전력($p_{n,m}^{u}$)의 합이 사용자 단말에 기지정된 최대 전송 전력(p_n^{max}) 이하이어야 하며, 사용자 단말의 로컬 CPU 사이클(f_n)이 기지정된 최소 CPU 사이클(f_n^{min})과 최대 CPU 사이클(f_n^{max}) 사이의 값을 가지고, 로컬 프로세싱을 포함한 다수의 RAT를 통한 오프로딩율의 총합은 1이다는 제약 조건을 만족하면서, eMBB 패킷에 대한 수학식(F_n^{e})이 최소가 되는 오프로딩율($o_{n,0}^{e}$, $o_{n,m}^{e}$)을 획득하는 사용자 단말의 오프로딩 방법.

청구항 20

제19항 있어서, 상기 URLLC 오프로딩율($1 - o_{n,0}^{u}$)을 획득하는 단계는

상기 URLLC 패킷에 대한 정규화된 에너지 소비량(E_n^{u})이 최소가 되는 URLLC 오프로딩율($1 - o_{n,0}^{u}$)과 함께 사용자 단말의 로컬 프로세싱을 위해 할당되어야 하는 로컬 CPU 사이클(f_n) 및 다수의 RAT 중 하나의 RAT를 통해 URLLC 패킷을 오프로딩하기 위한 URLLC 전송 전력($p_{n,m}^{u}$)을 유전 알고리즘을 이용하여 획득하고,

상기 eMBB 오프로딩율($o_{n,0}^{e}$, $o_{n,m}^{e}$)을 획득하는 단계는

상기 eMBB 패킷에 대한 수학식(F_n^{e})이 최소가 되는 오프로딩율($o_{n,0}^{e}$, $o_{n,m}^{e}$)과 함께 사용자 단말의 로컬 프로세싱을 위해 할당되어야 하는 로컬 CPU 사이클(f_n) 및 다수의 RAT를 통해 eMBB 패킷을 오프로딩하기 위한 eMBB 전송 전력($p_{n,m}^{e}$)을 유전 알고리즘을 이용하여 획득하는 사용자 단말의 오프로딩 방법.

발명의 설명

기 술 분 야

[0001] 본 발명은 동적 태스크 오프로딩을 수행하는 사용자 단말 및 이의 오프로딩 방법에 관한 것으로, 다중 액세스 엣지 컴퓨팅 환경에서 실시간 고복잡 IIoT 서비스를 위해 다중 무선 액세스 기술 기반 동적 태스크 오프로딩을 수행하는 사용자 단말 및 이의 오프로딩 방법에 관한 것이다.

배 경 기 술

[0002] 5G 기술은 다양한 산업용 사물 인터넷(Industrial Internet of Things: 이하 IIoT)을 효과적으로 지원하여 종단 간 사용자 서비스 품질(Quality of Service: 이하 QoS) 및 전체 프로세스의 안정성을 향상시킬 수 있다.

[0003] 특히 서로 다른 주파수의 통신 네트워크를 이용할 수 있는 다중 무선 액세스 기술(Multiple Radio Access Technology: 이하 다중 RAT)과 각 단말이 네트워크의 엣지에 분산된 다수의 에지 서버로 태스크를 전송하여 처리함으로써 하드웨어 성능 제한과 에너지 소모를 저감시키는 다중 액세스 엣지 컴퓨팅(Multi-access Edge Computing: 이하 MEC) 기술을 적용하여 QoS를 더욱 크게 향상시킬 수 있다.

[0004] 그러나 MEC와 다중 RAT를 적용할지라도 IIoT 어플리케이션은 매우 다양하고 각각 서로 상이한 요구 사항이 있기 때문에, 각 서비스의 유형에 따른 QoS를 동시에 모두 만족시키기는 매우 어렵다. 일 예로 공장 자동화, 프로세스 자동화 및 스마트 그리드 애플리케이션 등에 주로 이용되는 고신뢰 저지연 통신(ultra-reliable low latency communication: 이하 URLLC)의 경우 실시간성을 중요시하는 반면, 증강된 모바일 광대역(enhanced Mobile

BroadBand: 이하 eMBB)의 경우, 높은 대역폭을 통해 동영상이나 가상 현실(Virtual Reality: VR) 등과 같이 큰 패킷 사이즈의 대용량 데이터 프로세싱을 지원할 것을 요구한다. 따라서 지연 시간을 최소화할 것을 요구하는 URLLC와 대용량 데이터 프로세싱을 요구하는 eMBB의 QoS를 모두 만족시키는 것은 현실적으로 어렵다는 한계가 있다.

선행기술문헌

특허문헌

[0005] (특허문헌 0001) 한국 등록 특허 제10-1989249호 (2019.06.07 등록)

발명의 내용

해결하려는 과제

[0006] 본 발명의 목적은 서로 상이한 서비스 각각이 요구하는 QoS를 만족시키면서 성능을 최적화할 수 있는 동적 태스크 오프로딩을 수행하는 사용자 단말 및 이의 오프로딩 방법을 제공하는데 있다.

[0007] 본 발명의 다른 목적은 사용자 단말의 에너지 소비를 저감시킬 수 있는 동적 태스크 오프로딩을 수행하는 사용자 단말 및 이의 오프로딩 방법을 제공하는데 있다.

과제의 해결 수단

[0008] 상기 목적을 달성하기 위한 본 발명의 일 실시예에 따른 동적 태스크 오프로딩을 수행하는 사용자 단말은 MEC(Multi-access Edge Computing) 환경의 다중 RAT(Radio Access Technology) 네트워크에 포함되는 사용자 단말로서, 다중 RAT를 통해 MEC 서버와 통신을 수행하는 통신부; 상기 사용자 단말이 이용하는 서비스에 따라 처리되어야 하는 태스크 패킷이 임시 저장되는 로컬 큐; 상기 로컬 큐에 저장된 태스크 패킷 각각에 연산 자원인 로컬 CPU 사이클을 할당하여 처리하는 로컬 프로세서; 및 상기 서비스에 따라 획득된 태스크 패킷 중 상기 로컬 큐로 전달되어 상기 로컬 프로세서에서 처리되는 로컬 프로세싱율과 상기 통신부를 통해 상기 MEC 서버로 오프로딩되어 MEC 서버에서 처리되는 오프로딩율을 결정하는 오프로딩 스케쥴러를 포함하되, 상기 오프로딩 스케쥴러는 사용자 단말이 이용하는 서비스가 URLLC 서비스인 경우, 비트 단위로 URLLC 로컬 프로세싱율($o_{n,0}^{u}$)에 따른 에너지 소비($E_n^{u,L}$)와 URLLC 오프로딩율($1 - o_{n,0}^{u}$)에 따라 오프로딩되는 태스크 패킷인 URLLC 패킷의 전송 지연 시간($T_n^{u,\mathcal{T}}$)동안 단일 타임 슬롯(T_s) 단위의 전송 전력($p_{n,m}^{u}$)의 합으로 계산되는 정규화된 URLLC 에너지 소비량(E_n^{u})이 최소가 되는 URLLC 오프로딩율($1 - o_{n,0}^{u}$)을 유전 알고리즘을 이용하여 획득한다.

[0009] 상기 오프로딩 스케쥴러는 상기 URLLC 서비스의 태스크 각각 대한 태스크 패킷은 단일 URLLC 패킷으로 구성되고 로컬 프로세싱되거나 하나의 RAT를 통해 오프로딩되어, 상기 URLLC 패킷의 전송 지연($T_n^{u,\mathcal{T}}$)이 1 타임 슬롯(T_s) 이상이면, URLLC 패킷을 사용자 단말이 로컬 프로세싱하도록 URLLC 오프로딩율($1 - o_{n,0}^{u}$)을 0으로 획득할 수 있다.

[0010] 상기 오프로딩 스케쥴러는 상기 URLLC 패킷의 전송 지연($T_n^{u,\mathcal{T}}$)이 1 타임 슬롯(T_s) 미만이면, 수학식

$$F_n^u = \frac{o_{n,0}^u E_n^{u,\mathcal{L}} + (1-o_{n,0}^u) p_{n,m}^u T_n^{u,\mathcal{T}} T_s}{b_n^u}$$

[0011]

[0012] (여기서 F_n^{u}은 URLLC 패킷에 대한 정규화된 에너지 소비량($F_n^{u} = E_n^{u}$), b_n^{u} 는 URLLC 패킷 크기)이 최소가 되도록 하는 URLLC 오프로딩율($1 - o_{n,0}^{u}$)을 획득할 수 있다.

[0013] 상기 오프로딩 스케줄러는 상기 유전 알고리즘을 이용하여 URLLC 오프로딩율($1 - o_{n,0}^{u}$)을 획득함에 있어, URLLC 패킷의 전체 손실 확률(ε_n^{u})이 URLLC 서비스에 의해 기지정된 최대 손실 확률($\varepsilon_n^{u,max}$)이어야 하고, 단일 RAT를 통해 전송되는 URLLC 패킷의 전송 전력($p_{n,m}^{u}$)의 합이 사용자 단말에 기지정된 최대 전송 전력(p_n^{max}) 이하이어야 하며, 사용자 단말의 로컬 CPU 사이클(f_n)이 기지정된 최소 CPU 사이클(f_n^{min})과 최대 CPU 사이클(f_n^{max}) 사이의 값을 가지고, 로컬 프로세싱을 포함한 다수의 RAT를 통한 URLLC 오프로딩율의 총합은 1이다라는 제약 조건하에서 URLLC 패킷에 대한 정규화된 에너지 소비량(E_n^{u})이 최소가 되는 URLLC 오프로딩율($1 - o_{n,0}^{u}$)을 획득할 수 있다.

[0014] 상기 오프로딩 스케줄러는 상기 URLLC 패킷에 대한 정규화된 에너지 소비량(E_n^{u})이 최소가 되는 URLLC 오프로딩율($1 - o_{n,0}^{u}$)과 함께 사용자 단말의 로컬 프로세싱을 위해 할당되어야 하는 로컬 CPU 사이클(f_n) 및 다수의 RAT 중 하나의 RAT를 통해 URLLC 패킷을 오프로딩하기 위한 URLLC 전송 전력($p_{n,m}^{u}$)을 유전 알고리즘을 이용하여 획득할 수 있다.

[0015] 상기 오프로딩 스케줄러는 상기 사용자 단말이 이용하는 서비스가 eMBB 서비스인 경우, 비트 단위로 사용자 단말이 eMBB 태스크가 분할된 다수의 eMBB 패킷 중 직접 eMBB 패킷을 처리하는 eMBB 로컬 프로세싱율($o_{n,0}^{e}$)에 따른 eMBB 로컬 에너지 소비량($E_n^{e,L}$)과 다수의 RAT 각각에 대한 eMBB 오프로딩율($o_{n,m}^{e}$)에 따라 eMBB 패킷을 MEC 서버로 오프로딩하기 위한 전송 에너지 소비량($E_n^{e,\mathcal{T}}$)의 합으로 계산되는 정규화된 eMBB 에너지 소비량(E_n^{e})이 최소가 되는 eMBB 오프로딩율($o_{n,0}^{e}$, $o_{n,m}^{e}$)을 유전 알고리즘을 이용하여 획득할 수 있다.

[0016] 상기 오프로딩 스케줄러는 사용자 단말의 로컬 프로세싱에 따른 로컬 프로세싱 시간($T_{n,P}^{e,M}$)과 다수의 RAT 각각 통한 오프로딩 전송 지연 시간($T_{n,i}^{e,\mathcal{T}}$) 사이의 시간차 및 다수의 RAT를 통한 오프로딩 시의 각 RAT 사이의 전송 지연 시간차($T_{n,i}^{e,\mathcal{T}} - T_{n,j}^{e,\mathcal{T}}$)에 따라 계산되는 패널티($v_n$)를 상기 정규화된 eMBB 에너지 소비량($E_n^{e}$)에 가산한 결과가 최소가 되도록 eMBB 오프로딩율($o_{n,0}^{u}$, $o_{n,m}^{u}$)을 유전 알고리즘을 이용하여 획득할 수 있다.

[0017] 상기 오프로딩 스케줄러는 상기 MEC 서버가 다수의 사용자 단말 각각에서 오프로딩되는 태스크 패킷 중 URLLC 패킷과 eMBB 패킷을 서로 구분하여 저장하도록 독립적으로 구성되는 제1 태스크 큐 및 제2 태스크 큐를 구비하는 것으로 판단한다.

[0018] 상기 오프로딩 스케줄러는 상기 패널티(v_n)를 수학식

[0019]
$$\vartheta_n = \sum_{i,j \in M} \sqrt{(T_{n,\mathcal{P}}^{e,\mathcal{L}} - T_{n,i}^{e,\mathcal{T}})^2} + \sqrt{(T_{n,i}^{e,\mathcal{T}} - T_{n,j}^{e,\mathcal{T}})^2}$$

[0020] (여기서 $T_{n,P}^{e,L}$ 는 사용자 단말이 eMBB 패킷을 프로세싱하는데 소요되는 로컬 eMBB 프로세싱 시간, $T_{n,i}^{e,\mathcal{T}}$와 $T_{n,j}^{e,\mathcal{T}}$ 는 각각 다수의 RAT 중 i번째 및 j번째 RAT를 통한 오프로딩 전송 지연 시간)에 따라 계산할 수 있다.

[0021] 상기 오프로딩 스케줄러는 수학식

[0022]
$$F_n^e = \frac{k_e (f_n)^2 o_{n,0}^e c_n^e + \sum_{m \in M} \frac{p_{n,m}^e b_n^e o_{n,m}^e}{r_{n,m}^e}}{b_n^e} + \Omega \vartheta_n$$

[0023] (여기서 k_e 는 CPU의 처리 능력에 따라 달라지는 에너지 계수, f_n 은 사용자 단말의 로컬 CPU 사이클, c_n^{e} 는

MEC 서버가 오프로딩되어 제2 태스크 큐에 저장된 eMBB 패킷을 프로세싱하는데 요구되는 CPU 사이클, M은 RAT 개수, p_n^e 는 $p_{n,m}^e$은 m번째 RAT를 통해 오프로딩 시 사용자 단말의 eMBB 패킷 전송 전력, b_n^e는 eMBB 패킷 크기, $o_{n,m}^e$ 는 m번째 RAT를 통한 사용자 단말의 eMMB 오프로딩율, $r_{n,m}^e$ 는 eMBB 데이터율($r_{n,m}^e$), Ω는 패널티(v_n)의 반영 수준을 조절하기 위한 가중치이다.)이 최소가 되도록 하는 eMBB 오프로딩율($o_{n,0}^e$, $o_{n,m}^e$)을 획득할 수 있다.

[0024] 상기 오프로딩 스케쥴러는 상기 유전 알고리즘을 이용하여 eMBB 오프로딩율($o_{n,0}^e$, $o_{n,m}^e$)을 획득함에 있어, eMBB 태스크의 총 지연 시간(T_n^e)이 기지정된 eMBB 최대 허용 대기 시간($T_n^{e,max}$) 이하이어야 하고, 다수의 RAT를 통해 전송되는 eMBB 패킷의 전송 전력($p_{n,m}^u$)의 합이 사용자 단말에 기지정된 최대 전송 전력(p_n^{max}) 이하이어야 하며, 사용자 단말의 로컬 CPU 사이클(f_n)이 기지정된 최소 CPU 사이클(f_n^{min})과 최대 CPU 사이클(f_n^{max}) 사이의 값을 가지고, 로컬 프로세싱을 포함한 다수의 RAT를 통한 오프로딩율의 총합은 1이다라는 제약 조건하에서 eMBB 패킷에 대한 수학식(F_n^e)이 최소가 되는 오프로딩율($o_{n,0}^e$, $o_{n,m}^e$)을 획득할 수 있다.

[0025] 상기 오프로딩 스케쥴러는 상기 eMBB 패킷에 대한 수학식(F_n^e)이 최소가 되는 오프로딩율($o_{n,0}^e$, $o_{n,m}^e$)과 함께 사용자 단말의 로컬 프로세싱을 위해 할당되어야 하는 로컬 CPU 사이클(f_n) 및 다수의 RAT를 통해 eMBB 패킷을 오프로딩하기 위한 eMBB 전송 전력($p_{n,m}^e$)을 유전 알고리즘을 이용하여 획득할 수 있다.

[0026] 상기 목적을 달성하기 위한 본 발명의 다른 실시예에 따른 동적 태스크 오프로딩 방법은 MEC(Multi-access Edge Computing) 환경의 다중 RAT(Radio Access Technology) 네트워크에 포함되는 사용자 단말의 오프로딩 방법에 있어서, 다중 RAT를 통해 MEC 서버와 통신을 수행하여 MEC 서버와 다수의 RAT의 채널 상태 및 사용자 단말의 정보를 수집하는 단계; 상기 사용자 단말이 이용하는 서비스를 판별하는 단계; 사용자 단말이 이용하는 서비스가 URLLC 서비스이면, 상기 URLLC 서비스의 태스크 각각에 대응하는 단일 URLLC 패킷을 다수의 RAT 중 하나의 RAT를 통해 상기 MEC 서버로 오프로딩 시에 발생하는 전송 지연($T_n^{u,\mathcal{T}}$)이 1 타임 슬롯(T_s) 이상인지 판별하는 단계; 상기 URLLC 패킷의 전송 지연($T_n^{u,\mathcal{T}}$)이 1 타임 슬롯(T_s) 이상이면, URLLC 패킷을 사용자 단말이 로컬 프로세싱하도록 URLLC 오프로딩율($1 - o_{n,0}^u$)을 0으로 획득하는 단계; 및 상기 URLLC 패킷의 전송 지연($T_n^{u,\mathcal{T}}$)이 1 타임 슬롯(T_s) 미만이면, 유전 알고리즘을 이용하여 비트 단위로 URLLC 로컬 프로세싱율($o_{n,0}^u$)에 따른 에너지 소비($E_n^{u,L}$)와 URLLC 오프로딩율($1 - o_{n,0}^u$)에 따라 오프로딩되는 태스크 패킷인 URLLC 패킷의 전송 지연 시간($T_n^{u,\mathcal{T}}$)동안 단일 타임 슬롯(T_s) 단위의 전송 전력($p_{n,m}^u$)의 합으로 계산되는 정규화된 URLLC 에너지 소비량(E_n^u)이 최소가 되는 URLLC 오프로딩율($1 - o_{n,0}^u$)을 도출하는 단계를 포함한다.

발명의 효과

[0027] 따라서, 본 발명의 실시예에 따른 동적 태스크 오프로딩을 수행하는 사용자 단말 및 이의 오프로딩 방법은 요구사항이 서로 다른 다수의 서비스 각각의 특성과 요구 사항 및 단말의 에너지 소모를 고려하여 오프로딩 비율을 동적으로 결정함으로써 각 서비스에서 요구되는 QoS를 만족시키면서 에너지 소비를 저감하고 성능을 최적화할 수 있다.

도면의 간단한 설명

[0028] 도 1은 다중 RAT 기반 다중 MEC 환경 네트워크 오프로딩 시스템의 일 예를 나타낸다.

도 2는 본 발명의 일 실시예에 따른 사용자 단말의 개략적 구조를 나타낸다.

도 3은 본 발명의 일 실시예에 따른 사용자 단말 및 MEC 서버 큐 모델의 일 예를 나타낸다.

도 4는 본 발명의 일 실시예에 따른 사용자 단말의 오프로딩 방법을 나타낸다.

발명을 실시하기 위한 구체적인 내용

[0029] 본 발명과 본 발명의 동작상의 이점 및 본 발명의 실시에 의하여 달성되는 목적을 충분히 이해하기 위해서는 본 발명의 바람직한 실시예를 예시하는 첨부 도면 및 첨부 도면에 기재된 내용을 참조하여야만 한다.

[0030] 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.

[0031] 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.

[0032] 도 1은 다중 RAT 기반 다중 MEC 시스템의 일 예를 나타낸다.

[0033] 도 1에서는 하나의 매크로 기지국(Macro cell Base Station: 이하 MBS)(11) 및 다수의 소형 기지국(Small cell Base Station: 이하 SBS)(12)이 포함된 네트워크 시스템을 가정하여 도시하였다. 여기서 하나의 MBS(11)과 네트워크의 다수개의 SBS(12) 각각은 5G gNB 기능을 수행할 수 있는 것으로 가정하지만 이에 한정되지 않는다. 그리고 MBS(11) 및 다수의 SBS(12) 각각은 네트워크의 엣지로서 각각 MEC 서버를 구비하여 다중 MEC 시스템을 구성할 수 있으나, 여기서는 다수의 SBS(12)가 각각 MEC 서버를 구비하는 것으로 가정한다.

[0034] 다수의 SBS(12) 각각에 구비된 MEC 서버는 다수개의 사용자 단말(User Equipment: 이하 UE)(21, 22) 중 대응하는 셀 영역 내에 위치하는 UE(21, 22)로부터 전송되는 다양한 서비스의 태스크를 처리하는 오프로딩 기능을 제공할 수 있다. 여기서는 일 예로 QoS를 만족시키기 위한 요구 사항이 매우 상이한 URLLC 서비스 및 eMBB 서비스에 대해 오프로딩 기능을 제공하는 것으로 가정하여 설명한다.

[0035] 한편 MBS(11)의 매크로 셀 영역 내에는 다수개의 UE(20)가 위치할 수 있다. 도 1에서 다수개의 UE(20) 중 붉은색으로 표시된 제1 UE(21)는 URLLC 서비스를 이용하고, 파란색으로 표시된 제2 UE(22)는 eMBB 서비스를 이용하는 UE를 나타낸다. 그리고 다수개의 UE(20) 각각은 다중 RAT(Multi-RAT)에 기반하여 다수의 SBS(12)와 통신을 수행할 수 있다. 다중 RAT의 경우, 각 RAT가 서로 다른 주파수 대역을 사용하므로 상호 간섭은 발행하지 않는 것으로 가정한다. 일 예로 다중 RAT에는 2.4GHz의 WiFi, 1.8 ~ 2.3GHz의 LTE 및 3.7-4.2GHz의 중간 대역 5G가 이용될 수 있다.

[0036] 따라서 다수개의 UE(20) 각각은 자신이 수행해야하는 태스크를 다중 RAT에 기반하여, 대응하는 SBS(12)에 구비되는 MEC 서버로 전송하는 오프로딩을 수행할 수 있다.

[0037] 오프로딩 기법은 UE(22)의 부족한 태스크 처리 능력을 보완하고 에너지 소비를 저감시키기 위해 이용된다. 하지만 상기한 바와 같이, 각 서비스의 QoS를 만족시키기 요구되는 특성이 매우 상이하므로, 균일한 오프로딩 기법으로 모든 서비스의 QoS를 동시에 만족시키기 어렵다. 뿐만 아니라, 각각의 서비스에 대해 구분하여 개별적으로 오프로딩을 적용하더라도 서비스 특성에 따라서는 다양한 요인에 의해 MEC 서버로의 오프로딩이 오히려 QoS를 떨어뜨리는 결과를 초래할 수도 있다. 따라서 다수의 UE(20) 각각은 이용하는 각 서비스에 적합한 오프로딩율(offloading rate)을 설정할 수 있어야 한다. 특히 다수의 UE(20) 각각은 서비스에 따라 에너지 소비를 최소화하면서 태스크 처리 성능이 향상될 수 있도록 오프로딩율(o)을 동적으로 조절할 수 있어야 한다.

[0038] 도 2는 본 발명의 일 실시예에 따른 사용자 단말의 개략적 구조를 나타내고, 도 3은 본 발명의 일 실시예에 따른 사용자 단말 및 MEC 서버 큐 모델의 일 예를 나타낸다.

[0039] 도 2를 참조하면, 본 실시예에 따른 UE(20)은 통신부(210), 로컬 큐(220), 오프로딩 스케쥴러(230) 및 로컬 프로세서(240)를 포함할 수 있다.

[0040] 통신부(210)는 다수의 SBS(12)의 MEC 서버와 통신을 수행하여 각 서비스에 따른 태스크 패킷을 MEC 서버로 오프로딩할 수 있다.

[0041] 로컬 큐(220)는 UE(20)에서 생성된 서비스에 따른 태스크 패킷을 인가받아 임시 저장하고, 저장된 태스크 패킷을 지정된 순서에 따라 로컬 프로세서(240)로 전달한다. 이때, 각 UE(20)가 이용하는 서비스는 미리 지정될 수 있으며, 이에 로컬 큐(220)에는 지정된 서비스에 따른 태스크 패킷만이 저장된다. 여기서는 다수의 UE(20)가

각각 URLLC 서비스 또는 eMBB 서비스 중 하나를 이용하는 것으로 가정하였으므로, 로컬 큐(220)에는 URLLC 패킷 또는 eMBB 패킷이 저장될 수 있다.

[0042] 도 3에서는 설명의 편의를 위하여 다수의 UE(20) 각각의 로컬 큐(220)를 URLLC 서비스를 이용하는 제1 UE(21)의 로컬 큐(221)와 eMBB 서비스를 이용하는 제2 UE(22)의 로컬 큐(222)로 구분하여 도시하였다. 여기서 로컬 큐(220)는 FCFS(first come first serve) 모델에 따라 동작하는 것으로 가정한다. 즉 로컬 큐(220)는 우선 저장되는 태스크 패킷이 먼저 로컬 프로세서(240)로 전달되어 처리되도록 한다.

[0043] 한편, 로컬 프로세서(240)는 로컬 큐(220)에 저장된 태스크 패킷을 인가받아 프로세싱(processing)한다. 로컬 프로세서(240)는 CPU와 같은 연산 프로세서를 포함하여 구성되어 로컬 큐(220)에서 인가된 태스크 패킷을 순차적으로 처리할 수 있다.

[0044] 오프로딩 스케쥴러(230)는 이용하는 서비스에 따라 UE(20)에서 생성된 태스크 중 로컬 큐(220)에 전달되어 로컬 프로세서(240)에서 처리될 태스크 패킷과 MEC 서버로 오프로딩될 태스크 패킷을 구분한다. 본 실시예에서 오프로딩 스케쥴러(230)는 UE(20)가 이용하는 서비스에서 요구되는 QoS를 만족시키면서 UE(20)의 에너지 소비가 최소화될 수 있도록 오프로딩율(o)을 동적으로 결정하고 결정된 오프로딩율에 따라 오프로딩될 태스크 패킷을 판별할 수 있다.

[0045] 즉 다수의 UE(20) 각각은 지정된 방식에 따라 URLLC 태스크 또는 eMBB 태스크를 생성할 수 있으며, 생성된 URLLC 태스크 또는 eMBB 태스크의 패킷은 오프로딩 스케쥴러(230)에 의해 로컬 큐(221, 222)로 인가되거나 통신부(110)를 통해 MEC 서버로 오프로딩될 수 있다.

[0046] 이때 본 실시예에서는 MEC 서버에서 다수의 UE(20)에서 전송된 태스크 패킷을 인가받아 임시 저장하는 서비스 큐(120)가 인가되는 태스크 패킷의 서비스 종류에 따라 각각 구분하여 저장하도록 독립된 2개의 태스크 큐(121, 122)를 구비한다. 이에 오프로딩 스케쥴러(230)는 MEC 서버의 서비스 큐(120)가 인가되는 태스크 패킷의 서비스 종류에 따라 각각 구분하여 독립된 2개의 태스크 큐(121, 122)에 저장하는 것을 고려하여 오프로딩율(o)을 결정한다.

[0047] 일반적으로 MEC 서버의 서비스 큐(120)는 서비스에 따른 태스크 패킷을 구분하지 않는다. 그러나 서비스 큐 또한 FCFS 모델에 기반하여 저장된 데이터를 출력하므로, 서비스 큐(120)가 단일 큐로 구성되는 경우, 크기가 큰 eMBB 서비스의 태스크 패킷이 저장되면, 이후 오프로딩되어 저장되는 URLLC 서비스의 태스크 패킷은 요구되는 시간 조건을 만족시킬 수 없게 된다. 즉 오프로딩을 통한 URLLC 서비스의 QoS를 만족시킬 수 없게 되어, MEC 서버가 서로 다른 서비스에 대한 오프로딩을 수행할 수 없도록 한다. 이러한 문제를 방지하기 위해, 본 실시예에서는 MEC 서버의 서비스 큐(120)가 오프로딩을 지원하는 서비스 각각에 대응하여 독립적으로 구성되는 다수의 태스크 큐(121, 122)를 구비하는 것으로 가정한다.

[0048] 그리고 본 실시예에서 MEC 서버는 연산 프로세서인 CPU를 한정된 최대 CPU 사이클 자원(f_M^{max}) 범위 이내에서 CPU 사이클(CPU cycle) 단위로 자원을 동적으로 분할하여, 제1 및 제2 태스크 큐(121, 122)에 구분되어 저장된 URLLC 패킷과 eMBB 패킷에 동적으로 할당함으로써, 오프로드되어 제1 및 제2 태스크 큐(121, 122)의 대기열에서 대기중인 URLLC 패킷과 eMBB 패킷을 프로세싱할 수 있다고 가정한다. 따라서 MEC 서버의 동적으로 분할되는 CPU 사이클 중 오프로딩된 URLLC 패킷이 프로세싱되는 CPU 사이클을 나타내는 URLLC 서비스율(f_u)과 eMBB 패킷이 프로세싱되는 eMBB 서비스율(f_e)의 합은 MEC 서버의 사용 가능한 최대 CPU 사이클(f_M^{max}) 범위 이내($f_u + f_e \leq f_M^{max}$)이다.

[0049] 이하에서는 오프로딩 스케쥴러(230)가 오프로딩율을 최적화하는 방법을 구체적으로 설명한다.

[0050] 오프로딩 스케쥴러(230)는 UE측과 MEC 서버측의 관점을 구분하여 오프로딩 수행 여부에 따른 UE(20)의 에너지 소비를 분석하고, 이를 기초로 UE(20)의 에너지 소비가 최소가 되도록 오프로딩율(o)과 함께 오프로딩을 위한 태스크 전송 전력(p_n^{Ψ})과 각 UE(UE_n)의 연산 자원인 로컬 CPU 사이클(f_n)을 설정할 수 있다.

[0051] 여기서는 도 1과 같은 네트워크에 각각 태스크 동적 오프로딩 장치인 MEC 서버를 포함하는 K개의 SBS(12)가 배치되고, 각 SBS(12)의 셀 영역에 N개의 UE가 배치된 것으로 가정한다. 그리고 각 UE는 M개의 서로 다른 RAT를 이용하여 태스크 동적 오프로딩 장치와 통신을 수행하는 것으로 가정한다. 또한 여기서 시간은 5G NR 아키텍처에 따라 슬롯(slot) 단위로 이산화되고, 각 슬롯의 지속 시간은 T_s 로 표시되며, IIoT의 서비스(Ψ)에는 상기한

바와 같이 URLLC 서비스(u)와 eMBB 서비스(e)가 포함(ψ = {u, e})된다.

[0052] 그리고 N개의 UE 중 n번째 UE(UE_n)의 로컬 큐(220)는 이용하는 서비스(ψ)에 대응하는 태스크(J_n^{ψ} = λ_n^{ψ}, b_n^{ψ}, c_n^{ψ}, Q_n^{ψ}, n ∈ N)를 갖는다. 여기서 λ_n^{ψ}(task/slot)는 평균 태스크 도착율, b_n^{ψ}(bit/task)는 태스크의 패킷 크기, c_n^{ψ}(cycle/task)는 각 태스크를 처리하는 데 필요한 CPU 사이클이고, Q_n^{ψ}은 서비스의 QoS 제약이다.

[0053] 태스크 처리 CPU 사이클(c_n^{ψ})은 $c_n^{\psi} = b_n^{\psi} k_0$ 로 계산될 수 있으며, k_0 (cycle/byte)는 1 바이트를 처리하는 데 필요한 CPU 사이클 수를 나타낸다. QoS 제약(Q_n^{ψ})은 최대 허용 대기 시간($T_n^{\psi,max}$) 및 신뢰성($\varepsilon_n^{\psi,max}$)의 두가지 항목을 포함한다 (Q_n^{ψ} = {$T_n^{\psi,max}$, $\varepsilon_n^{\psi,max}$}. 또한 로컬 큐(220)에 태스크가 인가되는 평균 태스크 도착율(λ_n^{ψ})은 베르누이 분포(Bernoulli distribution)를 따르는 것으로 가정한다.

[0054] 한편, n번째 UE(UE_n)의 오프로딩율을 수학식 1과 같이 표현될 수 있다.

수학식 1

[0055]
$$o_{n,0}^{\psi} + \sum_{m \in M} o_{n,m}^{\psi} = 1$$

[0056] 여기서 $o_{n,0}^{\psi}$는 n번째 UE(UE_n)가 직접 태스크 패킷을 처리하는 로컬 프로세싱율(또는 로컬 오프로딩율)이고, $o_{n,m}^{\psi}$는 M개의 RAT 중 m번째(m = {1, 2, …, M}) RAT를 통해 n번째 UE(UE_n)가 태스크 패킷을 MEC 서버로 전달하는 오프로딩율을 나타낸다.

[0057] 즉 수학식 1은 태스크 패킷을 직접 처리하는 로컬 프로세싱율($o_{n,0}^{\psi}$)과 M개의 RAT 각각을 통해 MEC 서버로 태스크 패킷을 전송하여 처리하는 오프로딩율($o_{n,m}^{\psi}$)의 합은 태스크가 완전히 프로세싱되어야 하므로 1로 계산되어야 함을 나타낸다.

[0058] 그리고 URLLC 서비스의 경우, 상기한 바와 같이 URLLC 서비스에서는 실시간성을 중요시하여 태스크가 작은 고정된 크기(예를 들면 32byte)의 단일 패킷으로 구성되므로, 패킷의 분할 및 재구성으로 인한 추가 오버헤드가 발생되지 않도록, 패킷을 분할하지 않는다. 태스크가 분할될 수 없어 하나의 태스크가 하나의 패킷을 구성하므로, URLLC 서비스에서는 로컬 프로세싱율($o_{n,0}^{\psi}$) 또는 오프로딩율($o_{n,m}^{\psi}$) 중 하나가 1이 되고 나머지는 0이 되어야 한다.

[0059] 그리고 UE(UE_n)와 태스크 동적 오프로딩 장치 사이의 채널 상태가 좋지 않은 경우, 태스크 전송 시간을 고려하면 UE(UE_n)가 태스크를 오프로딩하지 않고 직접 처리하는 것이 더욱 효율적일 수 있다. 따라서 오프로딩 스케쥴러(230)는 URLLC 서비스에서 태스크가 UE(UE_n)에서 처리되거나, MEC 서버에 오프로딩되어 처리될 확률에 기반하여 오프로딩율을 최적화할 수 있다.

[0060] 반면, eMBB는 태스크의 크기가 크기 때문에, eMBB 태스크는 다수의 패킷으로 분할되어 M개의 RAT를 통해 분산 전송되어 지연 시간을 저감할 수 있다. 즉 eMBB 서비스에서는 태스크가 다수의 패킷으로 분할되어 자기 자신이 일부 패킷을 처리하고, 나머지 패킷은 M개의 RAT로 분산되어 MEC 서버로 오프로딩될 수 있다. 따라서 오프로딩 스케쥴러(230)는 eMBB 서비스에서 다중 RAT 기반 오프로딩을 통해 eMBB 서비스 지연 시간이 저감되도록 오프로딩율 최적화할 수 있다.

[0061] 여기서 오프로딩 스케쥴러(230)는 서비스별 QoS에 대한 요구 사항이 상이하므로, URLLC 서비스를 위한 오프로딩과 eMBB 서비스를 위한 오프로딩을 구분하여 분석함으로써, 더욱 신뢰성있는 최적의 오프로딩율을 획득할 수 있다.

[0062] 최적의 오프로딩율을 획득하기에 앞서 URLLC와 eMBB 서비스 각각에서 달성 가능한 데이터율을 우선 확인한다.

[0063] n번째 UE(UE_n)이 m번째 RAT를 이용하여 URLLC 패킷을 오프로딩하는 경우, 달성 가능한 URLLC 데이터율(data rate)($r_{n,m}^{u}$)은 대략적으로 수학식 2와 같이 계산될 수 있다.

수학식 2

[0064]
$$r_{n,m}^{u} \approx \frac{B_{n,m}^{u}}{\ln 2}\left[\ln\left(1+\frac{\alpha_{n,m}^{u} g_{n,m}^{u} p_{n,m}^{u}}{B_{n,m}^{u} N_0}\right)-\sqrt{\frac{V_{n,m}^{u}}{T_s B_{n,m}^{u}}} f_Q^{-1}(\varepsilon_n^{u,\mathcal{D}})\right]$$

[0065] 여기서 $B_{n,m}^{u}$ 는 m번째 RAT의 전송 대역폭으로 URLLC의 총 대역폭, $\alpha_{n,m}^{u}$ 및 $g_{n,m}^{u}$ 는 각각 대규모 및 소규모 채널 이득이며, $p_{n,m}^{u}$은 URLLC 전송 전력, N_0은 잡음 스펙트럼 밀도, f_Q^{-1}는 Q 함수(표준정규분포의 확률밀도함수(pdf)를 적분한 함수)의 역이다. 그리고 $\varepsilon_n^{u,D}$은 URLLC 패킷의 디코딩 오류 확률이고, $V_{n,m}^{u}$ 는 수학식을 간략화하기 위한 표현으로 $V_{n,m}^{u} = 1 - 1/(1+\frac{\alpha_{n,m}^{u} g_{n,m}^{u} p_{n,m}^{u}}{B_{n,m}^{u} N_0})^2$ 이다.

[0066] 수학식 2를 참조하면, URLLC 서비스의 데이터율은 새넌(Shannon)용량 공식에 따라 달성 가능한 데이터율에 URLLC의 짧은 패킷 길이로 인한 패킷 디코딩 오류 확률($\varepsilon_n^{u,D}$)을 추가로 반영하여 획득된다.

[0067] 그리고 n번째 UE(UE_n)이 m번째 RAT를 이용하여 eMBB 패킷을 오프로딩하는 경우, 달성 가능한 eMBB 데이터율($r_{n,m}^{e}$)은 수학식 3과 같이 계산될 수 있다.

수학식 3

[0068]
$$r_{n,m}^{e} = B_{n,m}^{e} \log\left(1+\frac{\alpha_{n,m}^{e} g_{n,m}^{e} p_{n,m}^{e}}{B_{n,m}^{e} N_0}\right)$$

[0069] 여기서 $B_{n,m}^{e}$ 는 m번째 RAT의 전송 대역폭이고, $\alpha_{n,m}^{e}$ 및 $g_{n,m}^{e}$ 는 각각 대규모 및 소규모 채널 이득이며, $p_{n,m}^{e}$은 eMBB 전송 전력을 나타낸다.

[0070] eMBB 서비스의 경우, URLLC 서비스에 비해 패킷 크기가 크기 때문에 디코딩 오류 확률($\varepsilon_n^{u,D}$)이 거의 없으므로 수학식 3에서는 수학식 2와 달리 패킷 디코딩 오류 확률($\varepsilon_n^{u,D}$)을 고려하지 않았다.

[0071] 한편, URLLC 서비스에서 태스크는 상기한 바와 같이 다수의 패킷으로 분할될 수 없어 단일 URLLC 패킷으로 구성되므로, UE에서 직접 프로세싱 되거나 태스크 동적 오프로딩 장치로 오프로딩되어 처리될 수 있다. 따라서 UE에서 직접 프로세싱되는 경우와 태스크 동적 오프로딩 장치로 오프로딩되어 처리되는 경우로 구분되어 분석될 수 있다.

[0072] 이에 n번째 UE(UE_n)의 URLLC 서비스의 오프로딩을 우선 분석한다.

[0073] UE에서 직접 처리되는 경우를 로컬 프로세싱(또는 로컬 오프로딩)이라 하며, 이때, n번째 UE(UE_n)가 URLLC 태스크를 처리하는 URLLC 로컬 프로세싱 시간($T_{n,P}^{u,L}$)은 수학식 4로 표현될 수 있다.

수학식 4

[0074]
$$T_{n,\mathcal{P}}^{u,\mathcal{L}} = \frac{c_n^u}{f_n}\ (slots)$$

[0075] 여기서 c_n^u은 UE(UE_n)가 URLLC 패킷을 처리하기 위해 요구되는 CPU 사이클이고, f_n은 UE(UE_n)의 로컬 CPU 사이클이며, L 은 UE(UE_n)의 로컬 동작임을 지시하기 위한 지시자이고, P는 프로세싱을 지시하기 위한 지시자이다.

[0076] UE(UE_n)의 로컬 CPU 사이클(f_n)이 고정된 패킷 서비스율(packet service rate)로 적용되는 경우, UE(UE_n)의 URLLC 로컬 큐(221)의 대기열은 Geo/D/1/FCFS 모델이 적용될 수 있다. Geo/D/1/FCFS 모델에 따르면 제1 큐(121)에서 대기열은 패킷 간격이 기하학적 분포(Geo)를 따르고, 결정된 서비스율(D)을 가지며 FCFS 모델에 따라 패킷이 출력된다.

[0077] URLLC 로컬 처리 시간($T_{n,P}^{u,L}$)과 유사하게 URLLC 로컬 큐(221)의 대기열에 의한 큐잉 지연 시간은 $T_{n,Q}^{u,L}$ 로 표현될 수 있다. 그리고, n번째 UE(UE_n)가 URLLC 태스크를 프로세싱하기 위한 URLLC 태스크 총 레이턴시($T_n^{u,L}$)는 큐잉 지연 시간($T_{n,Q}^{u,L}$)과 로컬 프로세싱 시간($T_{n,P}^{u,L}$)의 합으로 수학식 5와 같이 계산될 수 있다.

수학식 5

[0078]
$$T_n^{u,\mathcal{L}} = T_{n,\mathcal{P}}^{u,\mathcal{L}} + T_{n,\mathcal{Q}}^{u,\mathcal{L}} \le T_n^{u,max}$$

[0079] 여기서 $T_n^{u,max}$ 는 URLLC 서비스에서 QoS 를 만족시키기 위해 URLLC 패킷 처리에 요구되는 최대 허용 대기 시간을 의미한다.

[0080] URLLC의 QoS 를 만족시키기 위해서는 오류 확률과 시간 제약($T_n^{u,max}$)에 따른 신뢰성을 고려해야 하며, 이에 수학식 5로부터 로컬 프로세싱 시에 시간 제약($T_n^{u,max}$)을 초과하는 지연 위반 확률($\varepsilon_n^{u,L}$)을 수학식 6과 같이 계산할 수 있다.

수학식 6

[0081]
$$\varepsilon_n^{u,\mathcal{L}} = \Pr\left\{T_{n,\mathcal{Q}}^{u,\mathcal{L}} > T_n^{u,max} - T_{n,\mathcal{P}}^{u,\mathcal{L}}\right\} \le \varepsilon_n^{u,max}$$

[0082] 수학식 6으로부터 n번째 UE(UE_n)의 로컬 큐(220)에서의 큐잉 지연 시간($T_{n,Q}^{u,L}$)의 CCDF (Complementary Cumulative Distribution Function)는 수학식 7과 같이 표현될 수 있다.

수학식 7

[0083]
$$\begin{aligned}\Pr\left\{T_{n,\mathcal{Q}}^{u,\mathcal{L}} > i\right\} &= 1 - \Pr\left\{T_{n,\mathcal{Q}}^{u,\mathcal{L}} \le i\right\} \\ &= 1 - (1-q)^{-i-1}(1-\rho)\times\sum_{k=0}^{j}\left[q(1-q)^{D-1}\right]^k(-1)^k\binom{i+k-kD}{k}\end{aligned}$$

[0084] 여기서 i는 $(T_{n,P}^{u,max} - T_{n,P}^{u,L})$보다 큰 정수 중 가장 작은 값이고, $q = o_{n,0}^{u,L}$ 이며, $D = T_{n,P}^{u,L}$, $\rho = qD$ 이다.

[0085] 그리고 n번째 UE(UE_n)의 로컬 프로세싱에 따른 에너지 소비는 수학식 8로 계산될 수 있다.

수학식 8

[0086] $$E_n^{u,\mathcal{L}} = k_e(f_n)^2 c_n^u \ (J/packet)$$

[0087] 여기서 k_e 는 CPU의 처리 능력에 따라 달라지는 에너지 계수로서 일 예로 $k_e = 10^{-15}$이다.

[0088] 한편 n번째 UE(UE_n)가 URLLC 패킷을 MEC 서버로 오프로드 하는 경우, UE(UE_n)는 M개의 RAT 중 데이터율이 가장 높은, 즉 채널 상태가 가장 우수한 하나의 RAT 를 선택하여 URLLC 패킷을 전송한다. 따라서 URLLC 패킷의 전송 지연 시간($T_n^{u,\mathcal{T}}$)은 수학식 9로 계산될 수 있다.

수학식 9

[0089] $$T_n^{u,\mathcal{T}} = \frac{b_n^u}{\max_{m \in M} r_{n,m}^u T_s} \ (slots)$$

[0090] 여기서 $\mathcal{T}$ 는 전송 시간(transmission time)을 나타낸다.

[0091] URLLC 패킷이 무선 채널을 통해 전송되면, 패킷은 MEC 서버의 제1 태스크 큐(121)의 대기열에서 대기하게 된다. 이때 각각의 UE(UE_n)에서 MEC 서버로 도착하는 URLLC 패킷은 베르누이 분포를 따르지만 다수의 UE로부터 URLLC 패킷이 전송되므로, MEC 서버에 도착하는 전체 URLLC 패킷은 포아송 분포(Poisson distribution)를 따른다. 따라서 MEC 서버의 제1 태스크 큐(121)는 M/G/1 모델로 모델링될 수 있다. 그리고 MEC 서버에서 URLLC 패킷의 서버 프로세싱 시간($T_{n,P}^{u,M}$)은 수학식 10으로 표현될 수 있다.

수학식 10

[0092] $$T_{n,\mathcal{P}}^{u,\mathcal{M}} = \frac{c_n^u}{f_u} \ (slots)$$

[0093] MEC 서버에서 URLLC 패킷 총 지연 시간($T_n^{u,tot}$)은 제1 태스크 큐(121)에서의 큐잉 지연 시간($T_{n,Q}^{u,M}$)을 함께 고려하여 수학식 11로 표현될 수 있다.

수학식 11

[0094] $$T_n^{u,tot} = T_{n,\mathcal{P}}^{u,\mathcal{M}} + T_{n,\mathcal{Q}}^{u,\mathcal{M}} \leq T_n^{u,max}$$

[0095] 그리고 오프로딩에 따른 MEC의 오프로딩 시간 제약을 초과하는 URLLC 오프로딩 큐잉 지연 위반 확률($\varepsilon_n^{u,M}$)은

수학식 12로 표현될 수 있다.

수학식 12

[0096] $$\varepsilon_n^{u,\mathcal{M}} = \Pr\left\{T_{n,\mathcal{Q}}^{u,\mathcal{M}} > T_n^{u,max} + T_{n,\mathcal{P}}^{u,\mathcal{M}}\right\}$$

[0097] 수학식 12의 URLLC 오프로딩 지연 위반 확률($\varepsilon_n^{u,M}$)을 계산하기 위해서는 제1 태스크 큐(121)에 대기중인 패킷 수의 확률을 고려할 필요가 있다. 이를 위해, n번째 UE(UE_n)의 URLLC 패킷이 MEC 서버의 제1 태스크 큐(121)에 도달할 때 이미 제1 태스크 큐(121)에 저장되어 대기열에 대기중인 패킷의 수를 v^n 이라 가정한다.

[0098] 이 경우, $T_{n,\mathcal{Q}}^{u,\mathcal{M}}|_{v^n=v^*} > T_n^{u,max} + T_{n,\mathcal{P}}^{u,\mathcal{M}}$ 을 만족하는 최소 대기 패킷 수(v^*)가 있으면, URLLC의 지연 조건이 위반되는 것으로 볼 수 있다. 그러므로 큐잉 지연 위반 확률은 최소 대기 패킷 수(v^*) 이상인 MEC 서버의 제1 태스크 큐(121)의 대기 패킷 수 확률에 따라 감소된다.

[0099] 따라서 수학식 12의 URLLC 오프로딩 지연 위반 확률($\varepsilon_n^{u,M}$)은 수학식 13으로 근사될 수 있다.

수학식 13

[0100] $$\begin{aligned}\varepsilon_n^{u,\mathcal{M}} &= \Pr\left\{T_{n,\mathcal{Q}}^{u,\mathcal{M}} > T_{n,\mathcal{Q}}^{u,\mathcal{M}}|_{v^n=v^*}\right\} \\ &\approx \Pr\left\{v^n > v^*\right\}\end{aligned}$$

[0101] 한편 최소 대기 패킷 수(v^*)의 패킷이 제1 태스크 큐(121)의 대기열에 대기 중일 때, 대기열 지연은 대기 중인 각 패킷에 의한 프로세싱 대기 시간(W_n)과 현재 프로세싱되고 있는 패킷의 프로세싱 잔여 시간(R_n^u)으로 구분되어 계산될 수 있다. 따라서 대기 패킷 수(v^n)가 최소 대기 패킷 수(v^*)로 주어졌을 때 URLLC 대기열의 대기열 지연 시간($T_{n,Q}^{u,M}$)은 수학식 14로 나타날 수 있다.

수학식 14

[0102] $$T_{n,\mathcal{Q}}^{u,\mathcal{M}}|_{v^n=v^*} = \mathcal{W}_n + \mathcal{R}_n^u = \sum_{j=n-v^*}^{n-1} T_{j,\mathcal{P}}^{u,\mathcal{M}} + \mathcal{R}_n^u = \frac{v^* c_n^u}{f_u} + \mathcal{R}_n^u$$

[0103] 그리고 M/G/1 모델을 따르는 제1 태스크 큐(121)에 v개의 패킷이 대기중일 확률은 MEC 서버의 URLLC 대기열의 작업량(ρ_M)에 기반하여 수학식 15으로 계산될 수 있다.

수학식 15

[0104] $$\Pr\left\{v^n = v\right\} = (\rho_{\mathcal{M}})^v(1-\rho_{\mathcal{M}})$$

[0105] 여기서 $\sum_{v=0}^{\infty}\Pr\{v^n = v\} = 1$ 을 기준으로 할 때, 제1 태스크 큐(121)의 패킷 수가 v개 이상일 확률을 수학식

16으로 표현된다.

수학식 16

[0106] $$\Pr\{v^n > v\} = (\rho_{\mathcal{M}})^v$$

[0107] 수학식 16에서 MEC 서버의 URLLC 대기열의 작업량(ρ_M)은 수학식 17로 나타난다.

수학식 17

[0108] $$\rho_{\mathcal{M}} = \frac{\sum_{n \in N^u} (1 - o_{n,0}^u)\lambda_n^u c_n^u}{f_u}$$

[0109] 수학식 여기서 N^u는 MEC 서버의 URLLC 서비스 집합을 나타낸다.

[0110] 수학식 14 내지 16으로부터 URLLC 큐잉 지연 위반 확률($\varepsilon_n^{u,M}$)은 수학식 18과 같이 계산될 수 있다.

수학식 18

[0111] $$\Pr\left\{T_{n,\mathcal{Q}}^{u,\mathcal{M}} > \frac{v^* c_n^u}{f_u} + \mathcal{R}_n^u\right\} = \Pr\{v^n > v^*\} = (\rho_{\mathcal{M}})^{v^*}$$

[0112] 만일 $\frac{v^* c_n^u}{f_u} + \mathcal{R}_n^u = T_n^{u,max} + T_{n,\mathcal{P}}^{u,\mathcal{M}}$ 의 조건이 성립되면, URLLC 오프로딩 큐잉 지연 위반 확률($\varepsilon_n^{u,M}$)은 수학식 19과 같은 폐구조 형태로 유도될 수 있다.

수학식 19

[0113] $$\varepsilon_n^{u,\mathcal{M}} = (\rho_{\mathcal{M}})^{v^*} = \rho_{\mathcal{M}}\left(\frac{f_u(T_n^{u,max} + T_{n,\mathcal{P}}^{u,\mathcal{M}} - \mathcal{R}_n^u)}{c_n^u}\right)$$

[0114] 그리고 잔여 시간(R_n^u)은 현재 서비스 중인 패킷의 나머지를 프로세싱하는데 요구되는 시간 지연으로 $0 \le \mathcal{R}_n^u < T_{n,\mathcal{P}}^{u,\mathcal{M}} = \frac{c_n^u}{f_u}$ 를 만족한다. 따라서 URLLC 오프로딩 큐잉 지연 위반 확률($\varepsilon_n^{u,M}$)의 범위는 수학식 20으로 표현될 수 있다.

수학식 20

[0115] $$\rho_{\mathcal{M}}\left(\frac{f_u(T_n^{u,max} + T_{n,\mathcal{P}}^{u,\mathcal{M}})}{c_n^u}\right) \le \varepsilon_n^{u,\mathcal{M}} < \rho_{\mathcal{M}}\left(\frac{f_u(T_n^{u,max} + T_{n,\mathcal{P}}^{u,\mathcal{M}})}{c_n^u} - 1\right)$$

[0116] 여기서 상한은 현재 프로세싱되는 패킷이 프로세싱되기 시작한 직후인 최악의 상황을 고려한 대기 지연 위반 확

률을 나타내기 위해 사용된다.

[0117] 오프로딩된 URLLC 패킷에 대한 신뢰성은 수학식 2의 URLLC 패킷의 디코딩 오류 확률($\varepsilon_n^{u,D}$)과 수학식 20의 URLLC 오프로딩 큐잉 지연 위반 확률($\varepsilon_n^{u,M}$)에 의해 결정된다. 그리고 수학식 2의 URLLC 패킷의 디코딩 오류 확률($\varepsilon_n^{u,D}$)은 URLLC 패킷이 단일 타임 슬롯 구간 내에서 전송된다고 가정하여 계산할 수 있다.

[0118] URLLC 패킷이 단일 타임 슬롯 구간 내에서 전송되므로, $T_n^{u,\mathcal{T}} \leq T_s$ 이고, 따라서 하나의 패킷 전송 시에 $T_n^{u,\mathcal{T}} r_{n,m}^u = b_n^u \leq T_s r_{n,m}^u$ 또한 유지된다.

[0119] 최악의 시나리오로 $T_n^{u,\mathcal{T}} = T_s$ 인 경우를 고려하면, URLLC 패킷의 디코딩 오류 확률($\varepsilon_n^{u,D}$)의 상한은 수학식 21로 획득될 수 있다.

수학식 21

[0120]
$$\varepsilon_n^{u,\mathcal{D}} \approx f_Q\left(\sqrt{\frac{T_s B_{n,m}^u}{V_{n,m}^u}}\left[\ln\left(1+\frac{\alpha_{n,m}^u g_{n,m}^u p_{n,m}^u}{B_{n,m}^u N_0}\right)-\frac{b_n^u \ln 2}{T_s B_{n,m}^u}\right]\right)$$

[0121] 따라서 URLLC 패킷의 전체 손실 확률(ε_n^u)은 수학식 22와 같이 계산될 수 있다.

수학식 22

[0122]
$$\varepsilon_n^u = 1-(1-\varepsilon_n^{u,\mathcal{M}})(1-\varepsilon_n^{u,\mathcal{D}}) \leq \varepsilon_n^{u,max}$$

[0123] 수학식 22에 따르면, URLLC 패킷의 전체 손실 확률(ε_n^u)이 URLLC 서비스에 의해 기지정된 최대 손실 확률($\varepsilon_n^{u,max}$)이하이어야 한다.

[0124] URLLC 서비스에서 UE(UE_n)의 총 에너지 소비 모델은 UE(UE_n)의 로컬 프로세싱 및 전송 전력의 합으로 계산될 수 있다. 그리고 패킷의 로컬 프로세싱율 및 오프로딩율은 URLLC 평균 태스크 도착율(λ_n^u)과 URLLC 로컬 프로세싱율($o_{n,0}^u$) 및 URLLC 오프로딩율($1- o_{n,0}^u$)에 따라 각각 ($\lambda_n^u o_{n,0}^u$)과 ($\lambda_n^u(1- o_{n,0}^u)$)로 주어진다.

[0125] 따라서 URLLC 패킷에 대해 비트 단위로 예상되는 정규화된 URLLC 에너지 소비량(E_n^u)은 수학식 23과 같이 계산될 수 있다.

수학식 23

[0126]
$$E_n^u = \frac{\lambda_n^u o_{n,0}^u E_n^{u,\mathcal{L}} + \lambda_n^u (1-o_{n,0}^u) p_{n,m}^u T_n^{u,\mathcal{T}} T_s}{\lambda_n^u b_n^u}$$
$$= \frac{o_{n,0}^u E_n^{u,\mathcal{L}} + (1-o_{n,0}^u) p_{n,m}^u T_n^{u,\mathcal{T}} T_s}{b_n^u} \; (J/bit)$$

[0127] 한편, eMBB 서비스를 살펴보면, eMBB 서비스에서는 태스크를 다수의 eMBB 패킷으로 분할하고, 분할된 다수의

eMBB 패킷 중 일부를 UE(UE_n)이 직접 처리하고 나머지 패킷을 다수의 RAT를 통해 MEC 서버로 오프로딩하여 처리할 수 있다.

[0128] eMBB 서비스의 경우, QoS를 만족하기 위한 사용 제약 조건에 따라 UE(UE_n) 및 MEC 서버 각각의 안정성 조건을 통해 UE(UE_n) 및 MEC 서버 각각의 eMBB 작업량을 수학식 24 및 25와 같이 계산할 수 있다.

수학식 24

[0129]
$$\frac{o_{n,0}^{e}\lambda_{n}^{e}\overline{c_{n}^{e}}}{f_{n}} \leq 1$$

[0130] 여기서 $o_{n,0}^{e}$는 n번째 UE(UE_n)가 직접 태스크 패킷을 처리하는 로컬 프로세싱율, λ_n^e는 평균 eMBB 태스크 도착율, $\overline{c_n^e}$ 은 UE(UE_n)의 로컬 큐(222)에 대기중인 패킷을 프로세싱하는데 요구되는 UE(UE_n)의 평균 CPU 사이클을 나타낸다.

수학식 25

[0131]
$$\rho_{\mathcal{M}}^{e} = \frac{\sum_{n\in N^e}\lambda_{n}^{e}(1-o_{n,0}^{e})c_{n}^{e}}{f_e} \leq 1$$

[0132] 여기서 ρ_M^e 는 MEC 서버에서 제2 태스크 큐(122)의 eMBB 대기열의 작업량이고, c_n^e 는 MEC 서버가 제2 태스크 큐(122)에 대기중인 eMBB 패킷을 프로세싱하는데 요구되는 CPU 사이클, N^e 는 MEC의 eMBB 서비스 집합을 나타낸다.

[0133] 우선 n번째 UE(UE_n)의 eMBB 패킷에 대한 로컬 프로세싱 시간($T_{n,P}^{e,L}$)은 수학식 26과 같이 계산될 수 있다.

수학식 26

[0134]
$$T_{n,\mathcal{P}}^{e,\mathcal{L}} = \frac{o_{n,0}^{e}c_{n}^{e}}{f_n(Hz)}$$

[0135] 그리고 n번째 UE(UE_n)가 M개의 RAT를 통해 MEC 서버로 eMBB 패킷을 오프로딩하는 경우, m번째 RAT(RAT_m)을 통한 전송 지연($T_{n,m}^{e,\mathcal{T}}$)과 M개의 RAT를 통한 총 전송 지연($T_n^{e,\mathcal{T}}$)은 각각 수학식 27 및 28로 표현될 수 있다.

수학식 27

[0136]
$$T_{n,m}^{e,\mathcal{T}} = \frac{b_{n}^{e}o_{n,m}^{e}}{r_{n,m}^{e}}$$

[0137] 여기서 b_n^e 는 eMBB 태스크의 패킷 크기를 나타낸다.

수학식 28

$$T_n^{e,\mathcal{T}} = \max_{m \in M} \left[\frac{b_n^e o_{n,m}^e}{r_{n,m}^e} \right]$$

[0138]

[0139] eMBB 패킷은 M개의 RAT를 통해 병렬로 분산 전송될 수 있으므로, 수학식 28에 나타난 바와 같이, 개별 RAT를 통해 전송되는 전송 지연($T_{n,m}^{e,\mathcal{T}}$)의 최대값이 총 전송 지연($T_n^{e,\mathcal{T}}$)으로 계산된다.

[0140] 그리고 MEC 서버에서 n번째 UE(UE_n)에서 전송된 eMBB 패킷에 대한 프로세싱 지연 시간($T_{n,\mathcal{P}}^{e,M}$)은 수학식 29로 계산될 수 있다.

수학식 29

$$T_{n,\mathcal{P}}^{e,\mathcal{M}} = \frac{(1 - o_{n,0}^e) c_n^e}{f_e}$$

[0141]

[0142] URLLC 서비스의 경우, 모든 UE(22)에서 URLLC 패킷 크기(b_n^u)와 URLLC 패킷 처리 CPU 사이클 (c_n^u)이 동일하게 지정되지만, eMBB 서비스의 경우, 각 UE(22)가 상이한 eMBB 태스크의 패킷 크기(b_n^e)와 eMBB 패킷 처리 사이클 (c_n^e)을 가질 수 있다. 그리고 eMBB 서비스의 경우, 일반적으로 처리해야하는 태스크의 크기에 비해 출력되는 데이터의 크기가 매우 작아 무시할 수 있으므로, 오프로딩에 의해 태스크 동적 오프로딩 장치에서 처리된 태스크 처리 결과가 UE(22)에 전달되는 다운 링크 대기 시간은 무시될 수 있다.

[0143] 수학식 28 및 29로부터 eMBB 태스크 오프로딩에 따른 MEC 서버에서의 총 대기 시간($T_n^{e,M}$)은 수학식 30으로 계산될 수 있다.

수학식 30

$$T_n^{e,\mathcal{M}} = T_n^{e,\mathcal{T}} + T_{n,\mathcal{P}}^{e,\mathcal{M}}$$

[0144]

[0145] 상기한 바와 같이 eMBB 태스크 패킷은 n번째 UE(UE_n)와 MEC 서버에서 독립적으로 프로세싱 될 수 있으므로, eMBB 태스크의 총 지연 시간(T_n^e)은 수학식 31과 같이 표현될 수 있다.

수학식 31

$$T_n^e = \max \left\{ T_n^{e,\mathcal{L}}, T_n^{e,\mathcal{M}} \right\} \leq T_n^{e,max}$$

[0146]

[0147] 여기서 $T_n^{e,max}$ 은 eMBB 서비스의 QoS를 만족하기 위해 제한되는 eMBB 최대 허용 대기 시간을 나타낸다.

[0148] 정규화된 eMBB 에너지 소비량(E_n^e)은 단위 비트당 에너지 소비량으로 나타나므로, eMBB 태스크에 대한 정규화된 eMBB 에너지 소비량(E_n^e)은 UE(UE_n)이 로컬에서 eMBB 태스크 패킷을 프로세싱하는데 소비하는 eMBB 로컬 에너지 소비량($E_n^{e,L}$)과 eMBB 태스크 패킷을 MEC 서버로 오프로딩하기 위해 전송하는 전송 에너지 소비량($E_n^{e,\mathcal{T}}$)을 계산하여 합산하는 방식으로 수학식 32와 같이 계산될 수 있다.

수학식 32

$$E_n^{e,\mathcal{L}} = k_e (f_n)^2 o_{n,0}^e c_n^e$$

$$E_n^{e,\mathcal{T}} = \sum_{m \in M} \frac{p_{n,m}^e b_n^e o_{n,m}^e}{r_{n,m}^e}$$

$$E_n^e = \frac{E_n^{e,\mathcal{L}} + E_n^{e,\mathcal{T}}}{b_n^e} \; (J/bit)$$

[0149]

[0150] 본 발명에서 목적으로 하는 오프로딩 최적화는 이용하는 URLLC 또는 eMBB 서비스의 QoS를 만족하면서 UE(UE_n)의 에너지 소모를 최소화하는 것을 목적으로 하므로, UE(UE_n)의 에너지 소모 최소화 목적 함수를 수학식 33과 같이 표현할 수 있다.

수학식 33

[0151] $$\mathbb{E}_n(\mathbb{F}, \mathbb{O}, \mathbb{P}) = E_n^{\psi}$$

[0152] 여기서 ($\mathbb{F}, \mathbb{O}, \mathbb{P}$)는 로컬 CPU 사이클($\mathbb{F} = [f_n]$), 오프로딩율($\mathbb{O} = [o_{n,m}^{\psi}]_{m \in M}$), n 번째 UE($UE_n$)가 오프로딩을 위해 사용하는 전송 전력($\mathbb{P} = [p_{n,m}^{\psi}]_{m \in M}$)을 나타낸다.

[0153] 즉 서비스에 따른 UE(UE_n)의 에너지 소비량(E_n^{Ψ})이 최소가 되도록 하는 UE(UE_n)의 로컬 CPU 사이클(f_n), 오프로딩율($o_{n,m}^{\Psi}$) 및 전송 전력($p_{n,m}^{\Psi}$)을 획득해야 한다.

[0154] 수학식 33의 에너지 소모 최소화 목적 함수($\mathbb{E}_n(\mathbb{F}, \mathbb{O}, \mathbb{P})$)의 최적화는 서비스의 QoS를 만족시키기 위한 제약 조건(C1 ~ C6)을 고려하여 수학식 34로 정리될 수 있다.

수학식 34

$$\min_{\{\mathbb{F},\mathbb{O},\mathbb{P}\}_{m\in M}} \mathbb{E}_n(\mathbb{F},\mathbb{O},\mathbb{P})$$

$$\text{s.t } C1: \varepsilon_n^u \le \varepsilon_n^{u,max}$$
$$C2: T_n^e \le T_n^{e,max}$$
$$C3: T_n^{u,\mathcal{T}} \le T_s$$
$$C4: 0 \le \sum_{m\in M} p_{n,m}^{\psi} \le p_n^{max}$$
$$C5: f_n^{min} \le f_n \le f_n^{max}$$
$$C6: \sum_{m\in M} o_{n,m}^{\psi} = 1$$

[0155]

[0156] 수학식 34에서 제1 내지 제3 제약 조건(C1 ~ C3)은 UE의 로컬 프로세싱에 따른 서비스별 QoS 제약 조건을 나타내고, 제4 내지 제6 제약 조건(C4 ~ C6)은 로컬 CPU 사이클($\mathbb{F}=[f_n]$)와 오프로딩율($\mathbb{O}=[o_{n,m}^{\psi}]_{m\in M}$) 및 전송 전력($\mathbb{P}=[p_{n,m}^{\psi}]_{m\in M}$) 각각을 개별적으로 최적화하기 위한 QoS 제약이다.

[0157] 6가지 제약 조건 중 제1 및 제3 제약 조건(C1, C3)은 URLLC 서비스에 적용되어야 하는 제약 조건이고, 제2 제약 조건(C2)은 eMBB 서비스에 적용되어야 하는 제약 조건이며, 제4 내지 제6 제약 조건(C4 ~ C6)은 URLLC 서비스와 eMBB 서비스에 공통으로 적용되는 제약 조건이다.

[0158] 제1 제약 조건(C1)은 수학식 22에 따라 URLLC 패킷의 전체 손실 확률(ε_n^u)이 URLLC 서비스에 의해 기지정된 최대 손실 확률($\varepsilon_n^{u,max}$)이어야 한다는 제약 조건이고, 제2 제약 조건(C2)은 수학식 31에 따라 eMBB 태스크의 총 지연 시간(T_n^e)이 기지정된 eMBB 최대 허용 대기 시간($T_n^{e,max}$) 이하이어야 한다는 제약 조건이다. 그리고 제3 제약 조건(C3)은 수학식 9의 URLLC 패킷의 전송 지연 시간($T_n^{u,\mathcal{T}}$)이 단일 타임 슬롯(T_s) 이하이어야 한다는 조건이다.

[0159] 한편, 제4 내지 제6 제약 조건(C4 ~ C6)은 로컬 CPU 사이클($\mathbb{F}=[f_n]$)와 오프로딩율($\mathbb{O}=[o_{n,m}^{\psi}]_{m\in M}$) 및 전송 전력($\mathbb{P}=[p_{n,m}^{\psi}]_{m\in M}$) 각각을 개별적으로 최적화하기 위한 QoS 제약 조건이다. 제4 제약 조건(C4)은 단일 또는 다중 RAT를 통해 전송되는 각 서비스에 대한 태스크 패킷의 전송 전력($p_{n,m}^{\Psi}$)의 합이 UE(UE_n)에 기지정된 최대 전송 전력(p_n^{max}) 이하이어야 한다는 제약 조건이고, 제5 제약 조건(C5)은 UE(UE_n)의 로컬 CPU 사이클(f_n)이 기지정된 최소 로컬 CPU 사이클(f_n^{min})과 최대 로컬 CPU 사이클(f_n^{max}) 사이의 값을 가져야 한다는 제약 조건이며, 제6 제약 조건(C6)은 로컬 프로세싱에 따른 로컬 오프로딩을 포함한 다수의 RAT 전체를 통한 오프로딩율의 합은 1이 다라는 제약 조건이다.

[0160] 수학식 34의 최적화 문제는 다수의 제약 조건이 부가된 non-convex 함수로서 해가 없거나 해를 획득하기 매우 어려울 수 있다. 이에 본 실시예에서는 수학식 34의 최적화 문제를 유전 알고리즘(Genetic Algorithm: 이하 GA)를 이용하여 해결한다.

[0161] URLLC 서비스의 경우를 고려할 때, URLLC 서비스에서는 UE(UE_n)가 URLLC 패킷을 MEC 서버로 오프로딩하기 이전에 미리 채널 이득 및 대역폭과 같은 자원을 확인하여, URLLC 패킷이 1 타임 슬롯 이내에 전송 가능한 경우에만 오프로딩을 수행한다. 이는 전송 시간으로 1 타임 슬롯을 초과하게 되면, URLLC의 QoS를 충족할 수 없기 때문이다. 따라서 URLLC 패킷이 1 타임 슬롯 이내에 전송 불가능한 것으로 판별되면, UE(UE_n)는 URLLC 태스크의 모

든 패킷을 로컬 프로세싱한다. 즉 오프로딩을 수행하지 않는다. 한편, UE(UE_n)가 URLLC 패킷을 1 타임 슬롯 이내에 전송 가능하다고 판별한 경우, GA를 이용하여 에너지 소비를 최소화하는 최적의 로컬 CPU 사이클($\mathbb{F}$), 오프로딩율($\mathbb{O}$), 전송 전력($\mathbb{P}$)를 획득한다.

[0162] 반면, eMBB 서비스를 고려할 때, eMBB 태스크는 다수의 eMBB 패킷으로 분할되어 다수의 RAT를 통해 MEC 서버로 전송되어 UE(UE_n)와 MEC 서버에서 함께 eMBB 패킷이 처리될 수 있다. 이 경우 데이터율이 높은 RAT가 존재하면 UE(UE_n)는 다수 RAT를 통해 태스크를 많은 수의 패킷으로 분할하여 전송하는 반면, 채널 상태가 열악하여 데이터율이 낮으면 다수의 RAT 중 일부 개수의 RAT만을 이용하여 오프로딩을 수행하므로, UE(UE_n)의 로컬 프로세싱율($o_{n,0}^{e,L}$)이 높아진다. 그리고 eMBB 태스크의 지연 시간(T_n^e)은 로컬 eMBB 프로세싱 시간($T_{n,P}^{e,L}$)과 오프로딩 지연 시간($T_n^{e,M}$) 중 최대값에 의해 결정된다. 이때, eMBB 서비스에서는 일반적으로 처리해야하는 태스크의 크기에 비해 태스크 처리 결과로 출력되는 데이터의 크기가 매우 작으므로, 오프로딩에 의해 태스크 동적 오프로딩 장치에서 처리된 태스크 처리 결과가 UE(22)에 전달되는 다운 링크 대기 시간은 무시될 수 있다. 따라서 GA를 이용하여 UE(UE_n)의 에너지 소모를 최적화하는 경우, UE(UE_n)의 로컬 프로세싱에 따른 로컬 eMBB 프로세싱 시간($T_{n,P}^{e,L}$)과 다수의 RAT 각각 통한 오프로딩 전송 지연 시간($T_{n,i}^{e,\mathcal{T}}$) 사이의 시간차 및 다수의 RAT를 통한 오프로딩 시의 각 RAT 사이의 전송 지연 시간차($T_{n,i}^{e,\mathcal{T}} - T_{n,j}^{e,\mathcal{T}}$)를 패널티($v_n$)로서 추가로 반영할 필요가 있다. 여기서 UE(UE_n)의 eMBB 패킷의 다중 RAT 전송에 따른 패널티(v_n)와 패널티(v_n)를 반영한 에너지 소모 최소화 목적 함수($\mathbb{E}_n(\mathbb{F},\mathbb{O},\mathbb{P})$)는 수학식 35와 같이 계산될 수 있다.

수학식 35

$$\vartheta_n = \sum_{i,j\in M} \sqrt{(T_{n,\mathcal{P}}^{e,\mathcal{L}} - T_{n,i}^{e,\mathcal{T}})^2} + \sqrt{(T_{n,i}^{e,\mathcal{T}} - T_{n,j}^{e,\mathcal{T}})^2}$$

$$\min_{\{\mathbb{F},\mathbb{O},\mathbb{P}\}_{m\in M}} \mathbb{E}_n(\mathbb{F},\mathbb{O},\mathbb{P}) + \Omega\vartheta_n$$

[0163]

[0164] 상기한 수학식들을 참조하여, UE(UE_n)의 오프로딩 동작을 설명하면, 본 실시예에서 UE(UE_n)는 사용하는 IIoT 서비스가 URLLC 인지 eMBB 인지에 따라 최적화 방식이 구분된다.

[0165] 우선 UE(UE_n)가 URLLC 서비스를 이용하는 경우, 수학식 32의 6가지 제약 조건(C1 ~ C6) 중 제1 및 제3 내지 제6 제약 조건(C1, C3 ~ C6) 만족하면서 수학식 23의 정규화된 URLLC 에너지 소비량(E_n^u)이 최소가 되도록 최적화한다. 수학식 23의 URLLC 패킷에 대한 정규화된 URLLC 에너지 소비량(E_n^u)은 적합식으로 수학식 36로 표현된다.

수학식 36

$$F_n^u = \frac{o_{n,0}^u E_n^{u,\mathcal{L}} + (1-o_{n,0}^u) p_{n,m}^u T_n^{u,\mathcal{T}} T_s}{b_n^u}$$

[0166]

[0167] 수학식 36의 적합식(F_n^u)은 로컬 프로세싱율($o_{n,0}^u$)에 따른 에너지 소비($E_n^{u,L}$)와 오프로딩율($1-o_{n,0}^u$)에 따라 오프로딩되는 URLLC 패킷의 전송 지연 시간($T_n^{u,\mathcal{T}}$)동안 단일 타임 슬롯(T_s) 단위의 전송 전력($p_{n,m}^u$)의 합으로 계산되는 URLLC 패킷에 대해 비트 단위로 예상되는 정규화된 URLLC 에너지 소비량(E_n^u)을 의미한다.

[0168] 이때, 수학식 9로 계산되는 URLLC 패킷의 전송 지연($T_n^{u,\mathcal{T}}$)이 1 타임 슬롯보다 작다는 제3 제약 조건(C3)을 만족하는 경우, URLLC 패킷을 단일 타임 슬롯 구간 내에서 전송가능하므로, GA를 이용하여 수학식 36의 최적화 문제의 해를 탐색함으로써, 에너지 소모를 최소화하는 UE(UE_n)의 로컬 CPU 사이클($\mathbb{F}=[f_n]$), 오프로딩율($\mathbb{O}=[o_{n,m}^{\psi}]_{m\in M}$), 전송 전력($\mathbb{P}=[p_{n,m}^{\psi}]_{m\in M}$)을 획득한다.

[0169] 그러나 URLLC 패킷의 전송 지연($T_n^{u,\mathcal{T}}$)이 1 타임 슬롯 이상이면, 즉 제3 제약 조건(C3)을 위반하는 경우, URLLC의 QoS를 만족하는 오프로딩을 수행할 수 없으므로, URLLC 태스크의 모든 패킷을 UE(UE_n)가 로컬 프로세싱하도록 오프로딩율을 0으로 획득한다.

[0170] 한편, UE(UE_n)가 eMBB 서비스를 이용하는 경우, UE(UE_n)측에 대한 수학식 32의 제2 및 제4 내지 제6 제약 조건(C2, C4 ~ C6)을 만족하면서, 수학식 32의 eMBB 패킷에 대한 정규화된 eMBB 에너지 소비량(E_n^e)이 최소가 되도록 최적화한다. 다만, eMBB 서비스의 경우, 수학식 32의 에너지 소비량(E_n^e)에 eMBB 패킷의 다중 RAT 전송에 따른 수학식 35의 패널티(v_n)를 추가로 반영하여 정규화된 eMBB 에너지 소비량(E_n^e)이 최소가 되도록 최적화한다.

[0171] eMBB 패킷에 대한 정규화된 eMBB 에너지 소비량(E_n^e)에 패널티(v_n)를 반영한 적합식은 수학식 37로 나타난다.

수학식 37

[0172]
$$F_n^e = \frac{k_e (f_n)^2 o_{n,0}^e c_n^e + \sum_{m\in M} \frac{p_{n,m}^e b_n^e o_{n,m}^e}{r_{n,m}^e}}{b_n^e} + \Omega\vartheta_n$$

[0173] 여기서 Ω는 패널티(v_n)의 반영 수준을 조절하기 위한 가중치이다.

[0174] eMBB에서는 eMBB 태스크가 다수의 eMBB 패킷으로 분할되어, UE(UE_n)와 MEC 서버에서 분할 프로세싱 될 수 있으므로, URLLC와 달리 eMBB 패킷의 총 전송 지연($T_n^{e,\mathcal{T}}$)이 1 타임 슬롯보다 큰지 여부에 무관하게 GA를 이용하여 수학식 37의 최적화 문제의 해를 탐색함으로써, 에너지 소모를 최소화하는 UE(UE_n)의 로컬 CPU 사이클($\mathbb{F}=[f_n]$), 오프로딩율($\mathbb{O}=[o_{n,m}^{\psi}]_{m\in M}$), 전송 전력($\mathbb{P}=[p_{n,m}^{\psi}]_{m\in M}$)을 획득한다.

[0175] 결과적으로 본 실시예의 다중 RAT 기반 IIoT 서비스를 위한 동적 태스크 오프로딩을 수행하는 다수의 UE 각각은 사용하는 서비스가 URLLC 인지 eMBB 인지 여부에 따라 서로 상이한 QoS 조건을 만족시키면서 UE의 에너지 소모가 최소가 되도록 하는 오프로딩율을 획득할 수 있으며, 오프로딩율과 함께 로컬 CPU 사이클 및 전송 전력을 획득할 수 있다.

[0176] 도 4는 본 발명의 일 실시예에 따른 사용자 단말의 오프로딩 방법을 나타낸다.

[0177] 도 4를 참조하면, 본 실시예에 따른 오프로딩 방법은 우선 다수의 UE 각각이 다중 RAT 기반 MEC 네트워크와 UE에 대한 정보를 수집하여 획득한다(S10). 그리고 이용하는 IIoT 서비스를 확인하여, 확인된 IIoT 서비스가 URLLC 서비스인지 판별한다(S20).

[0178] 만일 URLLC 서비스인 것으로 판별되면, 수학식 34에 나타난 URLLC에서 요구하는 QoS를 만족하기 위한 제약 조건(C1, C3 ~ C6)을 확인한다(S30). 그리고 확인된 제약 조건(C1, C3 ~ C6) 중 제3 제약 조건(C3)인 수학식 9로 계산되는 URLLC 패킷의 전송 지연 시간($T_n^{u,\mathcal{T}}$)이 단일 타임 슬롯(T_s) 미만($T_n^{u,\mathcal{T}} < T_s$)인지 확인한다(S40). 즉 제3 제약 조건(C3)이 위배되는지 여부를 먼저 확인한다.

[0179] 만일 전송 지연 시간($T_n^{u,\mathcal{T}}$)이 단일 타임 슬롯(T_s) 이상이면, 태스크가 분할되지 않는 URLLC 서비스의 특성에

따라 오프로딩을 수행할 수 없으므로, URLLC 프로세스가 UE(UE_n) 자체에서 로컬 프로세싱 되도록 오프로딩율($o_{n,0}^{u}$ = 1)을 결정한다(S50).

[0180] 그러나 만일 전송 지연 시간($T_n^{u,\mathcal{T}}$)이 단일 타임 슬롯(T_s) 미만이면, URLLC 서비스에 대한 나머지 제약 조건(C1, C4 ~ C6)을 만족하면서, 수학식 36의 URLLC 패킷에 대한 정규화된 URLLC 에너지 소비량(E_n^{u})이 최소가 되도록 하는 UE(UE_n)의 오프로딩율($o_{n,m}^{u}$)을 GA를 이용하여 획득한다(S60). 이때, GA는 오프로딩율($o_{n,m}^{u}$)과 함께 로컬 CPU 사이클(f_n) 및 전송 전력($p_{n,m}^{u}$)을 획득한다. URLLC 태스크는 분할되지 않고 단일 패킷으로 구성되므로 UE(UE_n)는 미리 수집된 M개의 RAT의 채널 상태를 기반으로 채널 상태가 가장 양호한 하나의 RAT를 통해 MEC 서버로 오프로딩을 수행할 수 있다.

[0181] 한편, UE(UE_n)이 URLLC 서비스가 아닌 eMBB 서비스를 이용하는 것으로 판별되면, eMBB 서비스에서 요구하는 QoS를 만족하기 위한 제약 조건(C2, C4 ~ C6)을 확인한다(S70). 그리고 UE(UE_n)의 로컬 프로세싱에 따른 로컬 프로세싱 시간($T_{n,P}^{e,M}$)과 RAT를 통한 오프로딩 전송 지연($T_{n,i}^{e,\mathcal{T}}$) 및 다수의 RAT를 통한 오프로딩 시의 각 RAT의 전송 지연 시간차($T_{n,i}^{e,\mathcal{T}} - T_{n,j}^{e,\mathcal{T}}$)를 추가적으로 고려하기 위한 패널티 함수($v_n$)를 수학식 35에 따라 설정한다(S80).

[0182] 패널티 함수(v_n)가 설정되면, 제약 조건(C2, C4 ~ C6)을 만족하면서 수학식 32에 나타난 eMBB 패킷에 대한 정규화된 eMBB 에너지 소비량(E_n^{e})에 수학식 10의 패널티 함수(v_n)를 가산하여 수학식 37과 같이 획득된 적합식에 따라 에너지 소비량(E_n^{e})이 최소가 되도록 하는 UE(UE_n)의 로컬 프로세싱율($o_{n,0}^{e}$)과 M개의 RAT 각각에 대한 오프로딩율($o_{n,m}^{e}$)을 GA를 이용하여 획득한다(S90).

[0183] 본 발명에 따른 방법은 컴퓨터에서 실행시키기 위한 매체에 저장된 컴퓨터 프로그램으로 구현될 수 있다. 여기서 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.

[0184] 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.

[0185] 따라서, 본 발명의 진정한 기술적 보호 범위는 첨부된 청구범위의 기술적 사상에 의해 정해져야 할 것이다.

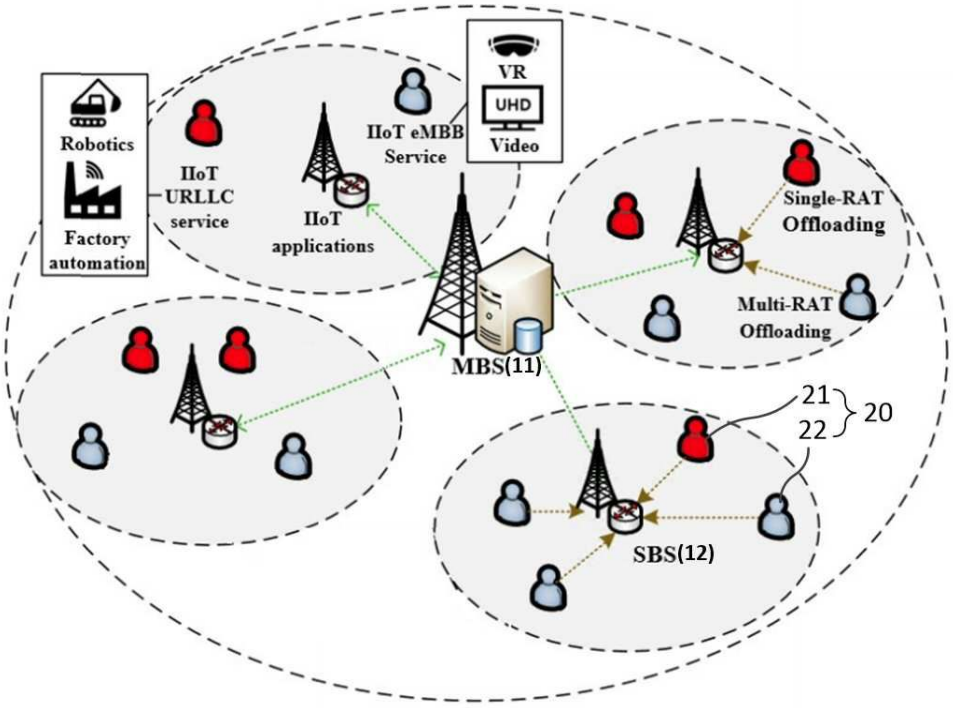
부호의 설명

[0186]

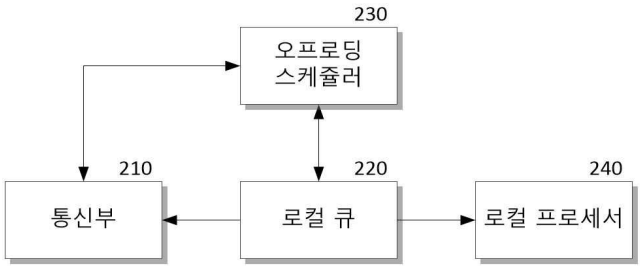
11: MBS
12: SBS
20: UE
210: 통신부
220: 로컬 큐
230: 오프로딩 스케쥴러
240: 로컬 프로세서

도면

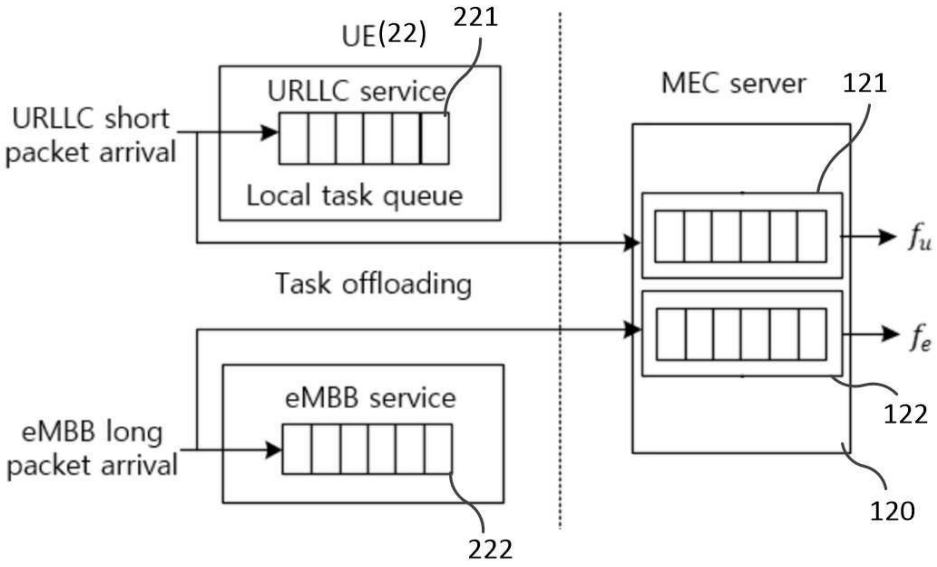
도면1


Robotics
Factory automation
IIoT URLLC service
IIoT applications
IIoT eMBB Service
VR
UHD
Video
Single-RAT Offloading
Multi-RAT Offloading
MBS(11)
21
22
20
SBS(12)

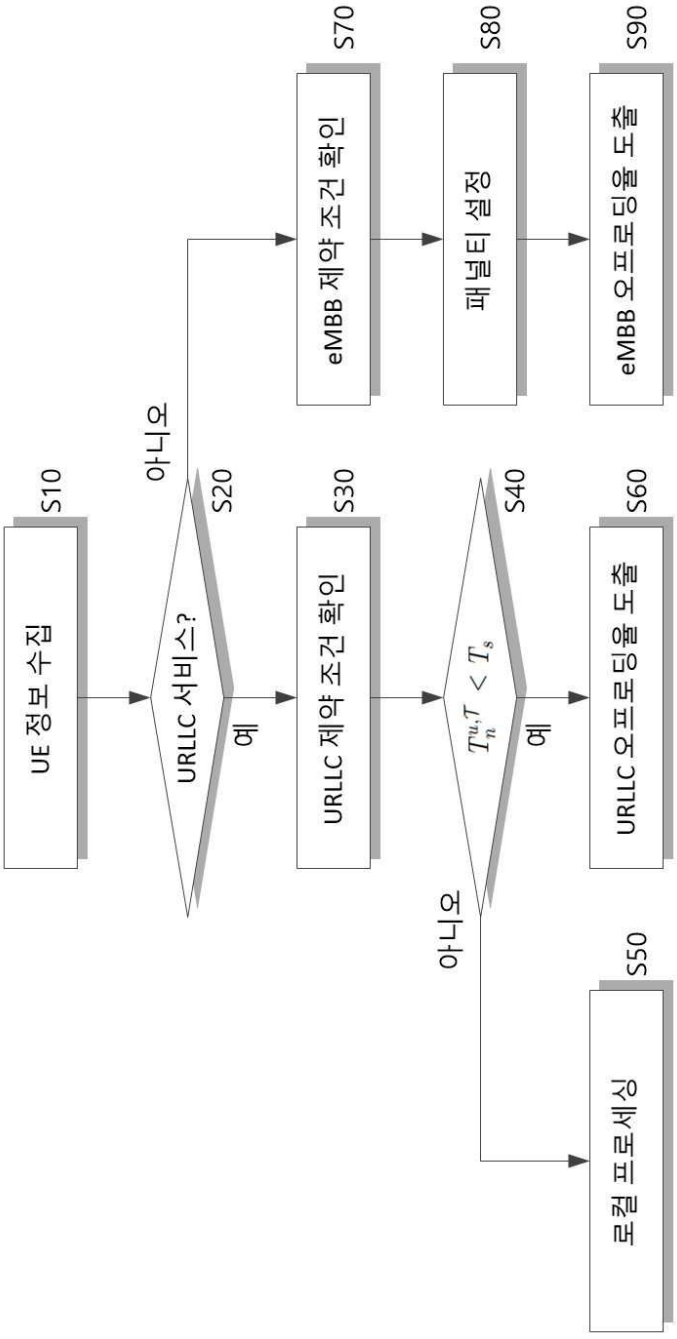
도면2


230
오프로딩 스케쥴러
210
통신부
220
로컬 큐
240
로컬 프로세서

도면3


UE(22)
221
URLLC service
URLLC short
packet arrival
Local task queue
MEC server
121
f_u
Task offloading
f_e
eMBB service
122
eMBB long
packet arrival
120
222

도면4


S10
UE 정보 수집
S20
URLLC 서비스?
예
아니오
S30
URLLC 제약 조건 확인
S40
$T_n^{u,T} < T_s$
예
아니오
S50
로컬 프로세싱
S60
URLLC 오프로딩율 도출
S70
eMBB 제약 조건 확인
S80
패널티 설정
S90
eMBB 오프로딩율 도출