

엣지 컴퓨팅과 차량: 미래를 위한 기회와 도전

정 성 아*

Edge Computing and Vehicles: Opportunities and Challenges for the Future

Seongah Jeong*

요 약

최근 5G/6G 등장과 함께 높은 컴퓨팅 능력을 요구하는 어플리케이션이 급증함에 따라, 한정된 처리 기능을 가진 사용자 기기에게 컴퓨팅 자원을 제공함으로써 원활한 서비스를 지원하는 엣지 컴퓨팅 기술이 많은 관심을 받고 있다. 특히, 사용자 기기에 가까이 엣지 노드를 설치하여 컴퓨팅 자원을 제공하는 엣지 컴퓨팅의 장점을 극대화하기 위해, 이동성이 자유로운 지상 차량 또는 무인 항공기 (UAV: Unmanned Aerial Vehicle)와 같은 항공 차량들이 함께 고려되어 활발히 연구가 진행되고 있는 실정이다. 본 논문에서는 차량을 고려한 대표적인 두 가지 엣지 컴퓨팅 시나리오, 차량 엣지 컴퓨팅 (VEC: Vehicular Edge Computing) 시스템과 모바일 엣지 컴퓨팅 (MEC: Mobile Edge Computing) 시스템에 대해 연구한다. 차량 엣지 컴퓨팅 시스템은 오프로딩 기기가 차량인 시스템으로써, 오프로딩을 제공하는 엣지 노드, 즉 RSU (Road Side Unit)가 차량의 높은 컴퓨팅 능력을 요구하는 차량 서비스의 원활한 지원을 제공한다. 또한, 모바일 엣지 시스템에서는 무인 항공기에 장착된 클라우드렛 (cloudlet)을 고려하여, 제한된 로컬 처리 기능을 가진 모바일 장치에 오프로딩을 제공하는 시나리오를 연구한다. 두 시스템 모두 한정된 배터리 기기, 즉 VEC 용 차량과 MEC용 모바일 기기가 높은 계산 능력을 필요로 하는 어플리케이션을 수행함에 따라, 많은 양의 에너지 소모를 야기한다. 이를 해결하기 위해, 본 논문에서는 RSU 또는 무인 항공기에 장착된 클라우드렛을 고려하여, 차량 혹은 모바일 기기의 총 에너지 소모량을 최소화하는 것을 목표로 하는 최적의 엣지 컴퓨팅 시스템을 연구한다. 제안 기술의 실용성을 높이고자, 직교 다중 접속 방식과 비직교 다중 접속 방식 모두를 고려하여, VEC향 최적의 비트 할당 및 오프 로딩 기술과 MEC향 최적의 비트 할당과 오프로딩 디자인 및 무인 항공기 궤적 기술을 연구한다. 시뮬레이션 결과를 통해, 로컬 컴퓨팅 대비, 차량과 모바일 장치의 상당한 에너지 절감을 확인할 수 있었고, 이를 통해 다양한 미래 엣지 컴퓨팅 기반 시스템의 발전 가능성을 확인하였다.

Key Words : edge computing, mobile edge computing (MEC), vehicular edge computing (VEC), internet of things (IoT), internet of vehicles (IoV), vehicle, unmanned aerial vehicle (UAV), ground vehicle

ABSTRACT

With the rapid increase of various applications that require the high computation resources, the edge computing has been recently considered as means to efficiently execute these applications. In order to maximize the performance of edge computing, the ground vehicles or Unmanned Aerial Vehicles (UAVs) have

* 이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. 2020R1G1A100626011).

◆ First Author and Corresponding Author : Kyungpook National University, School of Electronics Engineering, seongah@knu.ac.kr, 정희원

논문번호 : 202001-019-B-RU, Received January 18, 2021; Revised February 6, 2021; Accepted February 10, 2021

been employed. In this paper, two representative types of vehicle-based edge computing system are investigated such as Vehicular Edge Computing (VEC) and Mobile Edge Computing (MEC). In VEC systems, the VEC nodes, i.e., road side unit (RSUs), help to improve the high-complexity vehicular services and reduce the energy consumption of battery-limited vehicles via offloading. In MEC systems, a UAV-mounted cloudlet is adopted to offer the computation offloading opportunities to mobile devices with the limited local processing capabilities. For both systems, since the energy-limited vehicles for VEC and mobile devices for MEC are required the high computation capabilities, leading the high energy consumption, we aim at minimizing their total energy consumption by offloading to RSU or UAV-mounted cloudlet, respectively. To this end, both orthogonal and non-orthogonal multiple access are explored, whose optimal solutions are addressed by jointly optimizing the bit allocation and offloading for VEC, and the bit allocation, offloading and UAV trajectory for MEC.

1. 서론

스마트폰의 발명을 기점으로, 단일 혹은 다수의 센서를 장착한 수많은 스마트 기기들이 산업 전 분야에 걸쳐 폭발적으로 등장하기 시작하였다. 다양한 서비스 분야의 스마트 기기들이 용도별 목적에 맞게 실시간으로 주변 환경을 감지하고 모은 정보를 주변 기기들과 지속적으로 통신함에 따라, 상호 연결된 기기들로 이루어진 거대한 네트워크를 뜻하는 사물 인터넷 (IoT: Internet of Things)의 시대가 도래하게 되었다. 사물 인터넷의 출현은 컴퓨팅 시대의 대변혁을 일으키는 계기가 되었다^[1]. 헬스케어, 항공, 소셜 네트워킹, 운송 및 교통 등 사물 인터넷은 우리들의 일상에서 쉽게 찾아볼 수 있고, 늘어나는 스마트 기기들이 주고 받는 폭발적인 데이터를 처리하는 컴퓨팅 기술이 중요한 도전과제가 된 요즘이다.

엣지 컴퓨팅 (edge computing)은 컴퓨팅 환경에서 새로운 개념을 제시하였다. 모든 데이터를 하나의 장치에서 모두 처리하던 클라우드 컴퓨팅 (cloud computing) 서비스와 유틸리티를 최종 사용자에게 더 가깝게 제공하며, 빠른 처리와 빠른 어플리케이션 응답이 특징인 기술이다^[2]. 특히나, 긴급 재난전쟁 상황, 감시, 가상 현실 및 실시간 트래픽 모니터링과 같은 현재 개발된 인터넷 지원 어플리케이션들은 빠른 처리와 빠른 응답 시간이 필수적이다. 최종 사용자는 일반적으로 리소스가 제한된 모바일 장치에서 이러한 어플리케이션을 실행하고 핵심 서비스 및 처리를 클라우드 서버에서 수행된다. 하지만, 모바일 장치에서 클라우드 서비스를 활용하면 대기 시간이 길어지고, 모바일 장치의 배터리 소모가 빨라지며, 이동성 관련 문제 또한 발생하게 된다^[3]. 이를 위해, 엣지 컴퓨팅은 데이터 처리를 네트워크 엣지로 가져옴으로써 위의

문제들을 해결한다. 엣지 컴퓨팅의 장점인 이동성 지원, 초저 대기 시간 및 사용자와의 근접성 덕분에 다양한 미래 어플리케이션에 적합하다. 특히나, 이러한 엣지 컴퓨팅의 장점을 극대화시킬 수 있는, 혹은 장점이 필수적으로 요구되는 환경 요소로 이동성이 자유로운 차량이 최근 연구에서 많은 관심을 받고 있다.

본 논문에서는 차량이 고려되는 엣지 컴퓨팅 시스템을 크게 두 가지로 나누고, 각 시스템의 대표 시나리오를 살펴본다. 첫 번째로, 오프로딩 (offloading) 기기가 차량인 시나리오, 차량 엣지 컴퓨팅^[4] (VEC: Vehicular Edge Computing) 시스템을 고려한다. 두 번째로, 오프로딩된 기기, 즉 엣지 노드가 차량인 시나리오, 모바일 엣지 컴퓨팅^[6,17] (MEC: Mobile Edge Computing) 시스템을 고려한다. 본 논문에서의 차량은 지상에서 운용되는 차량뿐만이 아닌, 항공기나 드론 등의 항공 차량까지 포함하는 포괄적인 개념이다. 엣지 컴퓨팅 관련 현재까지의 연구들 중 엣지 컴퓨팅과 차량의 관계에 집중하여 심층적으로 다룬 연구는 아직까지 진행된 적이 없으므로, 본 논문을 통해 차량이 고려되는 대표적인 엣지 컴퓨팅 시스템, 즉 차량 엣지 컴퓨팅 시스템과 클라우드렛이 차량에 장착된 모바일 엣지 컴퓨팅 시스템을 살펴봄으로써, 차량이 고려되는 다양한 미래 엣지 컴퓨팅 기반 시스템 및 서비스에 대한 통찰력을 제공하고자 한다. 더 자세히는, 차량이 고려됨으로써 발생하는 두 엣지 컴퓨팅 시스템 설계의 공통점과 시스템 디자인 시 주의해야할 시나리오별 차이점에 대해 표 1에서 살펴보기로 한다.

표 1에서 보는 바와 같이, 두 시나리오 모두 차량을 고려한 엣지 컴퓨팅 시스템이기 때문에 비슷한 점도 많지만, 서로 다른 시스템 특징에 의해 시스템 설계 시 중요시 다뤄야 할 사항들이 있다. 더 자세히는, 차량 엣지 컴퓨팅 시스템은 오프로딩 기기인 차량이 한

표 1. 차량 엣지 컴퓨팅과 모바일 엣지 컴퓨팅 비교
Table 1. Comparison between VEC and MEC via UAV-mounted cloudlet

	VEC	MEC via UAV-mounted cloudlet
Offloading device	Vehicle - Limited battery - High mobility	Mobile device - Limited battery
Offloaded device (or Edge node)	Road side unit (RSU) - Sufficient battery - No mobility	UAV-mounted cloudlet - Limited battery - High mobility
Channel of interest	- Line-of-sight channel - Time-varying channel with Doppler effect	- Line-of-sight channel - Air-to-ground channel
Multiple access	- Orthogonal access - Non-orthogonal access	- Orthogonal access - Non-orthogonal access
Challenging issues	- Optimal bit allocation for commun. & computing - Optimal offloading ratio	- Optimal bit allocation for commun. & computing - Optimal offloading ratio - Optimal UAV trajectory

정된 배터리 기반의 빠른 이동성이 특징인 기기이지만, 엣지 노드는 도로 근처에 고정된 Road Side Unit (RSU)으로 풍부한 에너지 자원을 가진다. 이에 따라, 차량 엣지 컴퓨팅 시스템 관련 연구들은 차량과 RSU 사이 채널 모델로, Line-of-sight (LOS)가 지배적인 채널과 차량 이동성에 따른 도플러 효과를 고려한 채널에 집중하였다. 이에 반해, 무인 항공기 (UAV: Unmanned Aerial Vehicle)와 같이 비행 차량에 장착된 클라우드렛이 고려된 모바일 엣지 컴퓨팅 시스템의 경우, 한정된 배터리의 오프로딩 기기는 동일하지만, 엣지 노드인 무인 비행기 또한 한정된 배터리 기반의 이동성이 큰 기기를 고려한다. 특히, 사용자 접근 근성을 극대화하기 위해 3차원적으로 이동성이 자유로운 비행 차량을 고려하므로, 모바일 기기와 클라우드렛 사이 채널 모델로 LOS 지배적인 채널과 air-to-ground 채널이 활발히 연구 중이다. 두 시스템 모두 다수의 오프로딩 기기 지원이 목적이기 때문에, 직교 다중 접속 방식 (Orthogonal access)과 비직교 다중 접속 방식 (Non-orthogonal access)이 모두 고려 대상이다. 두 시스템 모두 한정된 배터리의 오프로딩 기기들의 전체 에너지 효율성 향상과 함께 최적의 리소스 (통신 비트 및 컴퓨팅 비트) 할당 기술 및 통신 환경에 따른 로컬 컴퓨팅과 오프로딩의 최적 비율이 중

요한 요소이다. 이에 더해, 무인 항공기를 고려한 모바일 엣지 컴퓨팅 시스템의 경우, 엣지 노드가 자유로운 이동성을 가질 수 있으므로, 오프로딩 성능을 극대화할 수 있는 무인 항공기의 최적 궤적 설계 또한 핵심 기술 요소이다.

본 논문에서는 차량 엣지 컴퓨팅과 모바일 엣지 컴퓨팅 시나리오에서 현재 이슈가 되고 있는 문제점들을 살펴보고, 최적화 문제를 형성하여, 솔루션을 연구한다. 특히, 이전의 차량 엣지 컴퓨팅^[10] 연구와 다르게, 직교/비직교 다중 접속 방식을 모두 고려하여 시스템 최적화를 제안하였고, 이전 모바일 엣지 컴퓨팅^[16,17] 연구 대비 추가적인 성능 이득을 위해 최적의 오프로딩 비율을 추가적으로 제안하였다.

II. 차량 엣지 컴퓨팅 시스템

이 장에서는 오프로딩 기기가 차량인 시나리오, 차량 엣지 컴퓨팅 시스템에 대해 연구한다. 사물 인터넷 적용 분야 중 가장 관심 받는 분야가 운송 분야인 만큼, 차량 인터넷 (IoV: Internet of Vehicles)을 주제로 많은 연구가 진행되고 있다^[4]. 특히, 자율주행, 비디오 스트리밍, 음성 인식 및 차량 엔터테인먼트 등 빠른 응답 속도와 높은 데이터 레이트와 컴퓨팅 기술을 요구하는 차량 어플리케이션이 증가함에 따라^[7], 차량과 멀리 떨어진 클라우드 서버에서 처리하는 클라우드 컴퓨팅 기술을 사용하는 것에 문제가 발생하기 시작하였다. 제한된 캐패시티 (capacity)의 백홀 링크 (backhaul link)를 통해 차량과 먼 클라우드 서버로 오프로딩을 시도하기 때문에, 많은 지연 시간을 초래하게 되었고, 이에 따라 응답 처리가 늦어지게 되었다. 이를 해결하기 위해, 차량 가까이 네트워크 엣지에서 오프로딩 서비스를 제공하는 차량 엣지 컴퓨팅 시스템이 해결책으로 등장하였다^[5].

차량 엣지 컴퓨팅 시스템은 차량과 가까운 엣지 노드에 오프로딩을 수행함으로써, 높은 컴퓨팅과 낮은 지연시간을 요구하는 차량 어플리케이션을 지원하는 데 효과적이다. 일반적으로, 모든 어플리케이션이 차량 내에서 수행되어, 전기차와 같은 에너지가 제한된 차량의 경우 막대한 에너지 소비가 발생하였다. 하지만, 차량 엣지 컴퓨팅 시스템의 경우, 원활한 에너지 공급과 높은 컴퓨팅 능력을 보장하는 엣지 노드에게 오프로딩을 시도하기 때문에, 차량 에너지 소비를 감소시켜 차량 배터리의 수명 연장이 가능하다^[4]. 이에 따라, 에너지 효율성이 고려된 오프로딩 전략 방법이 중요하다.

2.1 최근 연구 동향 분석

여기서는 최근 차량 엣지 컴퓨팅 시스템을 위한 오프로딩 기술들을 간략히 소개한다. 먼저, 차량의 이동성과 차량 어플리케이션 실행 시간을 고려하여, 전송 효율성을 높이는 오프로딩 기술 연구가 소개되었다^[4]. 또한, 분산적 방식으로 차량의 컴퓨팅 능력을 향상시키기 위해, 새로운 차량 엣지 컴퓨팅 시스템 프레임워크로써 자동 차량 엣지 (AVE: Autonomous Vehicular Edge) 기술이 제안되었다^[6]. 위의 소개된 연구들처럼, 대부분의 최근 차량 엣지 컴퓨팅 시스템 연구들은 지연 시간을 최소화하는 방향으로 연구가 많이 진행이 되어왔고, 소수 연구만이 에너지 효율성을 고려하여 연구가 진행되고 있다. 예를 들어, 제한된 배터리 성능을 가진 차량 내 사용자 기기들을 지원하기 위해 에너지 효율성이 고려된 차량 엣지 시스템^[7], ADMM (Alternating Direction Method of Multipliers) 기반 저 복잡도 오프로딩 솔루션^[8] 및 어플리케이션 요구 사항 지원을 위한 평균 송신 전력을 최소화하는 오프로딩 방법^[9] 등이 연구되었다. 하지만, 위의 연구들은 차량 내 사용자 기기 중심의 연구이거나 단일 차량만이 고려되었다. 따라서, 본 장에서는 실용성을 높이고자 다수 대의 차량 환경에서 모든 차량의 에너지 효율성이 고려된 오프로딩 기술을 연구한다.

2.2 차량 엣지 컴퓨팅 시스템 모델

본 장에서는 그림 1과 같이 다수 대의 차량이 존재하는 차량 엣지 컴퓨팅 시스템을 고려한다^[10]. 그림 1에서 보는 바와 같이, K 대의 차량과 M 개의 RSU로 구성된 차량 엣지 컴퓨팅 시스템으로써, RSU는 일방통행의 J 차선을 따라 설치되어있다고 가정한다. 이때, RSU간 간격은 d 이고, RSU의 커버리지 (coverage) 반지름은 r_{RSU} 라고 정의하면, RSU m 의 위치는 아래와 같이 표현될 수 있다.

$$p_m^r = (r_{RSU} + (m - 1)d, 0) \tag{1}$$

또한, K 대의 차량은 첫 번째 RSU의 커버리지에 각각 다른 시간 $t_k \in \{t_1, \dots, t_K\}$ 에 도달한다고 가정하고, j 차선의 차량의 속도는 $v_j \in \{v_1, \dots, v_J\}$ 로 정의한다^[11].

오프로딩 과정은 다음과 같다. 차량 k 가 가장 가까운 RSU에게 업링크를 통해 인풋 데이터를 전송한다. 데이터를 전송받은 RSU는 계산을 수행하고, 차량 k

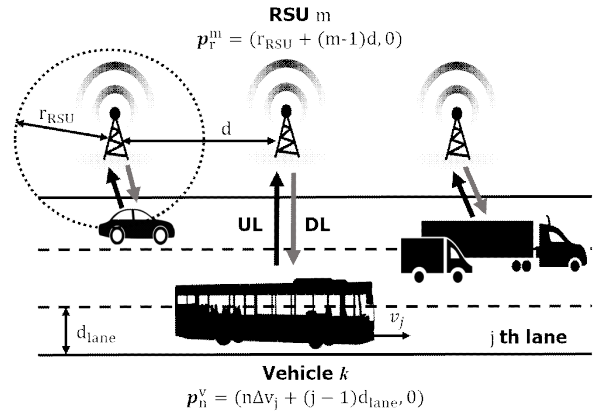


그림 1. 다수 대의 차량(오프로딩 기기)이 고려된 차량 엣지 컴퓨팅 시스템
Fig. 1. Illustration of the considered vehicular edge computing (VEC) system

에게 다운링크를 통해 아웃풋 데이터를 전송한다. 이때, 업링크 및 다운링크 통신은 FDD (Frequency Division Duplex)를 고려하여 업/다운링크 간 간섭은 존재하지 않고, 업/다운링크의 대역폭은 모두 B 이다. 그림 2에서 보는 바와 같이, 주어진 시간 T 는 길이가 Δ 인 N 개의 프레임으로 나뉘어져, $T = N\Delta$ 의 관계를 성립한다. 이 때, 프레임 길이 Δ 는 차량의 위치가 프레임 기간동안 일정하다고 가정할 수 있을 정도로 작게 산정된다. 다수대의 차량이 RSU 접속하는 방식에 따라, 그림 2에서 보는 바와 같이 2가지 시나리오를 고려해볼 수 있다. 직교 다중 접속 방식은 한 프레임을 오프로딩 기기가 나뉘어 사용하는 방식으로 오프로딩 수행기간동안 기기간 간섭이 발생하지 않는다. 이와 반대로, 비직교 다중 접속 방식은 오프로딩 기기가 하나의 프레임을 동시에 이용하는 방식으로 오프로딩 기간동안 기기간 간섭이 발생한다. 본 장에서는 직교 다중 접속 방식을 가정한다. 비직교 다중

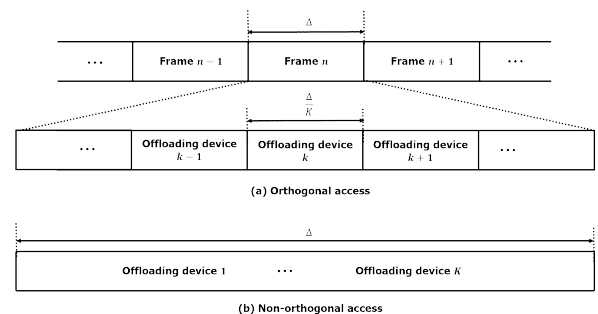


그림 2. 프레임 구조: (a) 직교 다중 접속 방식, (b) 비직교 다중 접속 방식
Fig. 2. Frame structure of the considered mobile cloud computing system: (a) Orthogonal access, (b) non-orthogonal access.

접속 방식은 모바일 엣지 컴퓨팅 시스템에서 소개된 방법으로 본 장의 내용이 쉽게 확장이 가능하다. 따라서, 각 차량은 Δ/K 기간동안 오프로딩을 위해 RSU와 통신한다. 위의 가정을 통해, j 번째 차선의 차량의 위치는 수식 (2)와 같이 정의된다.

$$p_n^v = (n\Delta v_j, (j-1)d_{lane}) \quad (2)$$

수식 (2)에서 $n \in \{1, \dots, N\}$, $j \in \{1, \dots, J\}$ 이고, d_{lane} 은 차선 넓이를 뜻한다. 차량과 RSU 사이 채널은 LOS가 지배적인 채널¹²⁾과 차량 이동성에 따른 도플러 효과를 고려한 채널을 가정할 수 있다. 본 논문에서는 차량 엣지 컴퓨팅 시스템에 대한 이해도 향상을 위해 LOS 링크를 가정한다. 이에 따라, n 번째 프레임에서 차량 k 와 RSU간 채널은 아래와 같이 정의된다.

$$h_{k,n}(p_n^v) = \frac{h_0}{\|p_n^v - p_{m_{\min}}^r\|^2 + H^2} \quad (3)$$

수식 (3)에서, $\|\cdot\|$ 는 norm-2 함수를 의미하고, m_{\min} 은 차량에서 가장 가까운 RSU의 인덱스를 뜻하며, H 는 RSU의 높이, h_0 는 $d = 1$ (meter)일 때, 1 W의 송신전력에 대한 수신전력의 크기이다. 또한, 채널 잡음은 평균은 0이고 power spectral density (psd)는 N_0 [dBm/Hz]인 additive White Gaussian noise (AWGN)을 가정한다.

차량 k 의 task는 인풋 데이터가 I_k 비트, 컴퓨팅 능력은 인풋 비트 당 C_k CPU 사이클, 아웃풋 데이터는 인풋 비트 당 O_k 비트를 아웃풋으로 계산한다고 가정한다. 대부분의 어플리케이션은 인풋 데이터의 크기가 아웃풋 데이터 크기에 비해 훨씬 크므로, $O_k \ll 1$ 의 관계를 만족하고, 모든 task는 데드라인 T 내에 완료되어야 한다.

1) 컴퓨팅 에너지 모델: CPU가 프리퀀시 f^i 로 동작할 때, 인풋 데이터가 l 인 차량 k 의 어플리케이션 컴퓨팅 에너지는 아래와 같다¹³⁾.

$$E_k^i(l, f^i) = \gamma^i C_k l (f^i)^2 \quad (4)$$

수식 (4)에서, 차량은 $i = v$ 이고, RSU는 $i = r$ 을

고려한다. 또한, f^i [CPU cycles/s]는 프로세서의 작동 프리퀀시를 뜻하며, γ^i 는 칩 구조와 관련된 프로세서의 effective switched capacitance를 나타낸다.

2) 통신 에너지 모델: 차량 k 가 Δ/K 슬롯 길이 동안 $L_{k,n}^q$ 비트를 통신한다고 가정하면, Shannon 이론에 의해 아래와 같은 관계식이 성립한다.

$$L_{k,n}^q = B \frac{\Delta}{K} \log_2 \left(1 + \frac{E_{k,n}^q(L_{k,n}^q, p_n^v) h_{k,n}(p_n^v)}{N_0 B \frac{\Delta}{K}} \right) \quad (5)$$

수식 (5)에서, $q = u$ 는 업링크를 의미하고, $q = d$ 는 다운링크를 의미한다¹⁴⁾. 수식 (5)를 통해, n 번째 프레임에서의 차량 k 의 통신 에너지 소모량을 아래와 같이 정의할 수 있다.

$$E_{k,n}^q(L_{k,n}^q, p_n^v) = \frac{N_0 B \Delta / K}{h_{k,n}(p_n^v)} \left(2^{\frac{L_{k,n}^q}{B \Delta / K}} - 1 \right) \quad (6)$$

수식 (6)에서 보는 바와 같이, 통신 에너지는 차량과 RSU간 거리로 정의되는 채널과 송신되는 비트에 영향을 받는다.

2.3 시스템 에너지 소모를 최소화하는 최적의 비트 할당 및 오프로딩 기술

여기서는 차량 엣지 컴퓨팅 시스템의 전체 에너지 소모량을 최소화하는 최적의 비트 할당 및 오프로딩 기술에 대해 연구한다. 첫 번째로, 벤치마크로 오프로딩이 고려되지 않았을 때의 경우인 로컬 컴퓨팅 시나리오에 대해 연구하고, 차량 엣지 컴퓨팅 시스템 에너지 소모를 최소화하는 최적 기술에 대해 연구한다.

로컬 컴퓨팅 에너지 소모량: 모든 어플리케이션이 차량 내에서 수행될 때의 전체 에너지를 계산한다. 데드라인 T 동안 I_k 비트 연산을 위해서, 차량 k 의 프리퀀시는 $f_k^v = C_k I_k / T$ 로 설정되어야 한다. 따라서, 로컬 컴퓨팅 시나리오의 전체 에너지 소모량은 수식 (7)과 같이 정의된다.

$$\sum_{k=1}^K E_k^v(I_k) = \sum_{k=1}^K \frac{\gamma_k^v C_k^3}{T^2} I_k^3 \quad (7)$$

수식 (7)에서 γ_k^v 는 차량 k 프로세서의 effective switched capacitance를 나타낸다.

차량 엣지 컴퓨팅 시스템의 에너지 소모량을 최소화하기 위해서는 채널 상황에 따라 로컬 컴퓨팅과 오프로딩을 적절히 선택하는 것이 최적일 것이다. 따라서, 차량 k 가 인풋 비트의 ρ_k 부분만 오프로딩을 고려하고, 나머지 부분은 로컬 컴퓨팅을 수행하는 시나리오가 에너지 효율성 측면에서 최적이다. 이 경우, 차량의 에너지 소모량은 오프로딩 수행을 위한 통신 에너지 소모량과 로컬 컴퓨팅을 위해 소모되는 컴퓨팅 에너지 소모량의 합으로 계산된다. 간단히, 오프로딩 과정을 프레임별로 분석하면 다음과 같다. 예를 들어, 첫 번째 프레임에서 차량 k 가 $L_{k,1}^u$ 비트를 RSU에게 업링크를 통해 전송하고, 이 때 컴퓨팅 비트나 다운링크 통신 비트는 존재하지 않는다 (즉, $l_{k,1}^c = L_{k,1}^d = 0$). 두 번째 프레임에서는 차량 k 가 $L_{k,2}^u$ 비트를 RSU에게 업링크를 통해 전송하고, RSU는 $l_{k,2}^c \leq L_{k,1}^u$ 비트를 계산한다. 이 때, 다운링크 통신 비트는 존재하지 않는다 (즉, $L_{k,2}^d = 0$). 세 번째 프레임에서는 차량 k 가 $L_{k,3}^u$ 비트를 RSU에게 업링크를 통해 전송하고, RSU는 $l_{k,3}^c \leq L_{k,1}^u + L_{k,2}^u - l_{k,2}^c$ 비트를 계산하며, RSU는 $L_{k,3}^d \leq \kappa_k l_{k,2}^c$ 의 아웃풋 데이터를 차량 k 에게 전송한다. 이러한 방식으로 N 프레임동안 오프로딩 과정을 수행한다. 이에 따라, 차량 엣지 컴퓨팅 시스템 에너지를 최소화하는 최적의 비트 할당 $\{L_{k,n}^u\}, \{l_{k,n}^c\}, \{L_{k,n}^d\}$ 및 최적의 오프로딩 $\{\rho_k\}$ 디자인 문제는 아래와 같이 표현된다.

수식 (8)에서 조건 1, 2, 3은 오프로딩 데이터의 완료를 보장하고, 조건 4는 오프로딩 비율 조건을 나타낸다. 또한, I_{max} 는 프레임당 동일 RSU로 전송될 수 있는 업링크 최대 전송 비트를 의미한다. 전체 차량 에너지 소모량을 최소화하는 방향으로 오프로딩 할당량과 비트 할당량이 수식 (8)을 통해 산출된다. 수식 (8)은 컨벡스 프로그래밍 (convex programming)이므로 표준 컨벡스 최적 기술 방법, 즉 generic solver (CVX MOSEK)나 Lagrangian dual variable을 사용하여 해결이 가능하다.

2.4 모의 실험 및 분석

본 장에서는 차량 엣지 컴퓨팅 시스템 최적화 기술

$$\begin{aligned}
 & \min_{\{L_{k,n}^u\}, \{l_{k,n}^c\}, \{L_{k,n}^d\}, \{\rho_k\}} \sum_{k=1}^K \sum_{n=1}^{N-2} E_{k,n}^u (L_{k,n}^u) \\
 & \quad + \sum_{k=1}^K E_k^v ((1 - \rho_k) I_k) \\
 & s.t. \quad \sum_{n=1}^{N-2} L_{k,n}^u = \rho_k I_k, \\
 & \quad \sum_{n=1}^{N-2} l_{k,n+1}^c = \rho_k I_k, \\
 & \quad \sum_{n=1}^{N-2} L_{k,n+2}^d = O_k \rho_k I_k, \\
 & \quad 0 \leq \rho_k \leq 1, \\
 & \quad \sum_{i=1}^n l_{k,i+1}^c \leq \sum_{i=1}^n L_{k,i}^u, \\
 & \quad \sum_{i=1}^n L_{k,i+2}^d \leq O_k \sum_{i=1}^n l_{k,i+1}^c, \\
 & \quad \sum_{k=1}^K L_{k,n}^u \leq I_{max}, \\
 & \quad L_{k,n}^u, l_{k,n}^c,
 \end{aligned} \tag{8}$$

의 성능을 분석한다. 시뮬레이션에서는 $K = 10$ 대의 차량과 $J = 3$ 차선을 고려하였다. 또한 가장 빠른 차량 속도 v_{max} 를 고려하여, $M = \lceil v_{max} T - r_{RSU}/d + 1/2 \rceil$ 개의 RSU를 가정하였다. 모든 차량의 데드라인을 T 로 고려하고, 첫 번째 RSU 커버리지에 도착하는 차량은 랜덤으로 정해진다고 가정하였다. 그 외 시스템 파라미터는 [10]을 참고한다. 기술의 성능 비교를 통해, 세 가지 레퍼런스를 고려한다. 첫 번째로 차량 내에서 모든 컴퓨팅을 수행하는 로컬 컴퓨팅 (local computing), 두 번째로 모든 차량이 오프로딩을 고려하되, 각 프레임에서 업링크 및 다운링크 통신 비트와 RSU에서의 컴퓨팅 비트가 모두 동일한 complete offloading with equal bit allocation, 세 번째로 모든 차량 어플리케이션의 오프로딩을 고려하되, 업링크 및 다운링크 통신 비트와 RSU에서의 컴퓨팅 비트가 최적화된 complete offloading with optimal bit allocation (수식 (8)에서 모든 k 에 대하여, $\rho_k = 1$ 인 경우)로, 수식 (8)의 최적 오프로딩 및 비트 할당 기술 (optimal offloading and bit allocation)과 비교한다.

그림 3에서 보는 바와 같이, 데드라인이 짧아질수록, 최적화 기술의 성능 효과가 극대화됨을 알 수 있다. 또한, 최적 비트가 할당되지 않은 equal bit

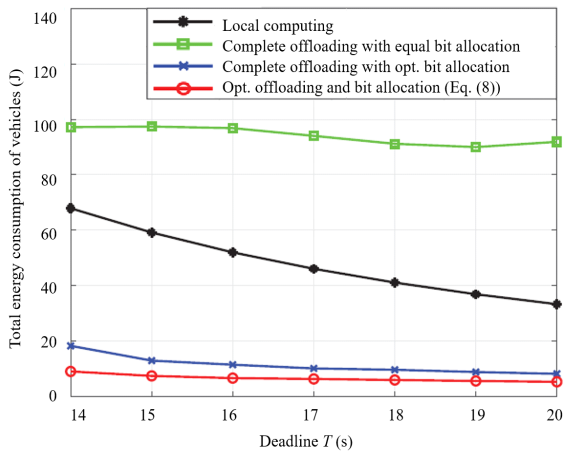


그림 3. 차량 엡지 컴퓨팅 시스템에서 데드라인 T 에 따른 전체 차량 에너지 소모량 비교
 Fig. 3. Total energy consumption of the vehicles as a function of the deadline T

allocation을 고려하면 로컬 컴퓨팅보다 에너지 소모가 현저하게 높은 것을 관찰할 수 있다. 이는 오프로딩을 위한 통신 에너지가 채널 환경에 따라 변화될 수 있으므로, 통신 환경에 따른 적절한 오프로딩 사용이 중요함을 알 수 있다.

III. 모바일 엡지 컴퓨팅 시스템

이 장에서는 오프로딩된 기기, 즉 엡지 노드가 차량인 시나리오인 모바일 엡지 컴퓨팅 시스템에 대해 연구한다. 더 자세히는, 엡지 컴퓨팅의 장점을 극대화하기 위해 공중을 비행하는 무인 비행 차량 혹은 무인 항공기 기반 엡지 노드를 고려한다. 특히, 개발 도상국이나 교외지역 혹은, 긴급 재난 환경 및 군사 환경 등과 같은 제한적인 인프라의 무선 액세스 포인트 (wireless access point) 환경에서 무인 항공기를 이용한다면 삼차원적 이동성의 자유도와 함께 무선 시스템의 커버리지를 효과적으로 확장할 수 있다^[15-18]. 하지만, 배터리가 한정된 무인 항공기의 무선 시스템 적용 시, 시스템 디자인 목적에 맞게 무인 항공기의 올바른 경로와 위치 설정이 중요한 이슈이다.

3.1 최근 연구 동향 분석

여기서는 최근 무인 항공기 기반 모바일 엡지 컴퓨팅 시스템 기술들을 간략히 소개한다. 먼저, 무인 항공기의 효과적인 모바일 엡지 컴퓨팅 시스템 적용을 위해, 무선 시스템 환경에서의 무인 항공기 관련 연구를 살펴보기로 한다.

무선 기기간 통신을 위해 무인 항공기가 릴레이

(relay)로 동작하는 기술들이 연구되었다^[19-22]. 자세히는, 무인 항공기가 decode-store-and-forward 방식의 릴레이 통신을 지원할 때, 무인 항공기의 한계 속력과 출발 및 도착 지점의 제한 사항 등의 이동성 요구 사항을 고려하여 시스템 스루풋 (throughput)을 최대화하는 기술이 연구되었다^[19]. 또한, 소스 노드와 타겟 사이를 비행하는 무인 항공기 기반 릴레이를 고려하는 애드 hoc 네트워크 (ad hoc network) 환경에서, 메시지 삭제율 (message drop rate)을 최소화하는 무인 항공기 비행 경로 최적화 기술^[20,21] 및 지상 기지국과 사용자 기기 간 업링크 통신 환경에서 전체 데이터 레이트를 최대화하는 최적의 무인 항공기 릴레이 배치 기술^[22]이 연구되었다. 이와 함께, 무인 항공기가 기지국 형태로 투입되어, 다운링크 통신 환경을 지원하는 기술^[23,24]들이 소개되었다. 다운링크 환경에서 지상 기기들의 통신을 원활히 지원하며 송신 전력을 최소화하는 무인 항공기의 최적 높이^[23] 및 전체 사용자 데이터 레이트를 최대화하는 최적의 무인 항공기 위치 기술들이 연구되었다^[24]. 위의 연구에서 보는 바와 같이, 무인 항공기의 제한적 배터리 및 자유로운 이동성으로 인해 무선 시스템 적용 시, 이 두 특징에 유의하여 도입됨을 확인할 수 있다. 이는 모바일 엡지 컴퓨팅 시스템에도 동일하게 적용된다^[15-17]. 구체적으로, 무선 인프라가 부족한 환경에서 사물 인지 혹은 증강 현실 (AR: augmented reality) 등과 같이 높은 계산 복잡도를 요구하는 어플리케이션의 오프로딩이 필요한 경우, 무인 항공기 기반 엡지 노드가 효과적으로 이용될 수 있다. 다만, 이전 장에서 소개되었던 차량 엡지 시스템에서의 엡지 노드인 RSU는 풍부한 에너지 자원을 보유하고 고정 노드였다면, 무인 항공기 기반 엡지 노드는 제한적인 배터리와 이동 노드이기 때문에 기존 무인 항공기 적용 무선 시스템과 같이 시스템 디자인 목표에 맞게 최적의 활용 기술이 요구된다. 따라서, 본 장에서는 무인 항공기 기반 엡지 노드의 활용성을 극대화하고자 모바일 엡지 컴퓨팅 환경에서 사용자 기기의 에너지 효율성이 고려된 오프로딩 기술을 연구한다.

3.2 모바일 엡지 컴퓨팅 시스템 모델

본 장에서는 그림 4와 같이 무인 항공기가 엡지 노드로 고려된 모바일 엡지 컴퓨팅 시스템을 연구한다^[15-17]. 그림 4에서 보는 바와 같이, 무인 항공기가 K 개의 지상 모바일 기기에게 오프로딩을 제공하는 엡지 노드로 동작하는 상황에서, 모바일 기기의 에너지 소비를 최소화하는 최적의 비트 할당과 오프로딩 디

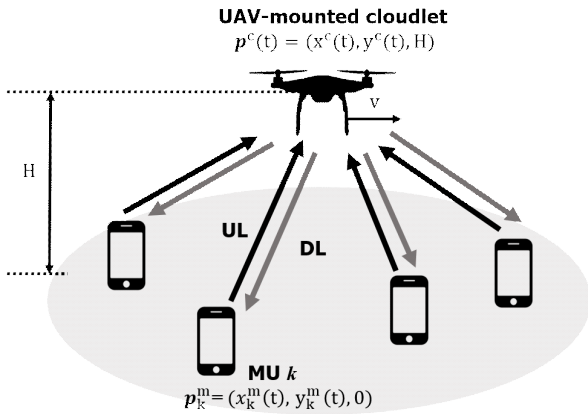


그림 4. 무인 비행 차량 혹은 무인 항공기 (엣지 노드)가 고려된 모바일 엣지 컴퓨팅 시스템
 Fig. 4. Illustration of the considered mobile edge computing (MEC) system based on a UAV-mounted cloudlet

자인 및 무인 항공기 제적 기술을 연구한다. 오프로딩 과정은 다음과 같다. 먼저, 모바일 기기 k 가 무인 항공기에게 업링크를 통해 인풋 데이터를 전달한다. 무인 항공기에 장착된 클라우드렛이 전달된 데이터를 계산한 후, 계산이 완료된 아웃풋 데이터를 무인 항공기가 다시 모바일 기기 k 에게 전송한다. 이 때, 업링크 및 다운링크 통신은 FDD (Frequency Division Duplex)를 고려하여 업/다운링크 간 간섭은 존재하지 않는다. 또한, 차량 엣지 컴퓨팅 시스템과 비슷하게 다수 개의 모바일 기기가 무인 항공기에 접속하는 방식에 따라, 그림 2에서 보는 바와 같이 2가지 시나리오, 즉 직교 다중 접속 방식과 비직교 다중 접속 방식을 고려할 수 있다. 본 장에서는 5G/6G에서 활발히 연구되고 있는 NOMA 시나리오와 같이, 모바일 기기가 동시에 함께 접속하는 비직교 다중 접속 방식을 소개한다. 직교 다중 접속 방식은 섹션 II의 차량 엣지 컴퓨팅 시스템에서 소개된 방법으로 본 장의 내용이 쉽게 확장이 가능하다. 비직교 다중 접속 방식을 가정하였기 때문에, 업링크 및 다운링크 시 한 프레임 시간인 Δ 기간 동안 모든 모바일 기기는 통신을 수행한다. 본 논문에서는 통신 및 정보 이론에 따라, 원치 않는 신호로부터 파생된 간섭 신호는 모두 additive noise로 취급한다^[25]. 모바일 기기의 task는 인풋 데이터가 I_k 비트, 컴퓨팅 능력은 인풋 비트 당 C_k CPU 사이클, 아웃풋 데이터는 인풋 비트 당 O_k 비트를 아웃풋으로 계산한다고 가정하고, 모든 task는 데드라인 T 내에 완료되어야 한다. 차량 엣지 컴퓨팅 시스템처럼 그림 2에서 보는 바와 같이, 주어진 시간 T 는 길이가 Δ 인 N 개의 프레임으로 나뉘어져, $T = N\Delta$

의 관계를 성립한다. 이 때, 프레임 길이 Δ 는 무인 항공기의 위치가 프레임 기간동안 일정하다고 가정할 수 있을 정도로 작게 선정된다.

그림 4에서 보는 바와 같이, 무인 항공기 기반 모바일 엣지 컴퓨팅 시스템 설계를 위해 3차원 데카르트 좌표계 (Cartesian coordinate system)을 고려한다. 모든 모바일 기기는 지상, 즉 xy 평면상에 존재한다고 가정하고, 모바일 기기 k 의 위치는 $p_k^m = (x_k^m, y_k^m, 0)$ 로 나타낸다. 본 논문에서는 무인 항공기에 대한 원활한 이해를 위해, 편의상 지상과 평행하게 비행하는 무인 항공기의 궤도를 고려한다. 하지만 본 논문에서 연구한 기술에 air-to-ground 채널을 반영하여 자유로운 고도 비행의 무인 항공기를 고려한 시나리오로의 확장이 쉽게 가능하다. 따라서, H 의 높이로 평행 비행을 하는 무인 항공기의 궤도는 $p^c(t) = (x^c(t), y^c(t), H)$ 로 표현할 수 있다. 하지만, 일정한 고도의 비행을 고려하므로, xy 평면에 프로젝션한 궤적으로 간단히 표현할 수 있고, 그림 2에서의 이산 시간 가정으로 인해 n 번째 프레임에서의 무인 항공기 위치는 $p_n^c = (x_n^c, y_n^c)$ 로 정의할 수 있다. 또한, 무인 항공기의 이착륙 지점 및 비행 능력 요구 사항을 위해, 이착륙 지점 및 최대 속력을 $p_I^c = (x_I^c, y_I^c)$, $p_F^c = (x_F^c, y_F^c)$, v_{max} 로 정의한다. 최대 속력 제한에 따라, n 번째 프레임에서의 무인 항공기 속도 $v_n^c = \frac{p_{n+1}^c - p_n^c}{\Delta}$ 는 $\|v_n^c\| \leq v_{max}$ 조건을 항상 만족해야 한다. 모바일 기기와 무인 비행기 간 채널은 Line-of-sight (LOS)가 지배적인 채널^[15-17]과 고도에 따라 영향을 받는 air-to-ground 채널을 고려할 수 있다. 본 논문에서는 평행 고도 비행의 무인 항공기를 고려하므로, LOS가 지배적인 채널을 고려하도록 한다. 이에 따라, n 번째 프레임에서 모바일 기기 k 와 무인 항공기 간 채널은 아래와 같이 정의된다.

$$g_{k,n}(p_n^c) = \frac{g_0}{(x_n^c - x_k^m)^2 + (y_n^c - y_k^m)^2 + H^2} \quad (9)$$

수식 (9)에서 g_0 는 $d = 1$ (meter)일 때, 1 W의 송신전력에 대한 수신전력의 크기이다. 또한, 채널 잡음은 평균은 0이고 power spectral density (psd)는 N_0

[dBm/Hz]인 additive White Gaussian noise (AWGN) 을 가정한다.

1) 컴퓨팅 에너지 모델: 모바일 기기 k 의 어플리케이션 컴퓨팅을 위한 에너지 모델은 섹션 II-2-1)의 차량 엣지 시스템에서 소개된 컴퓨팅 에너지 모델 (4)와 동일하게 고려한다. CPU가 프리퀀시 f^i 로 동작할 때, 인풋 데이터가 l 인 모바일 기기 k 의 어플리케이션 컴퓨팅 에너지는 수식 (4)와 동일하게 적용된다. 이 때, 차량은 $i = m$ 이고, 클라우드렛은 $i = c$ 을 고려한다.

2) 통신 에너지 모델: 본 논문에서는 비직교 다중 접속을 고려하므로, 모든 모바일 기기가 동시에 Δ 기간 동안 통신을 수행한다. 모바일 기기 k 와 클라우드렛이 n 번째 프레임에서 각각 $L_{k,n}^u$ 비트와 $L_{k,n}^d$ 비트를 업링크 및 다운링크를 통해 통신한다고 가정하면, Shannon 이론에 의해 업링크 에너지 소모량 $E_{k,n}^u(L_n^u, p_n^c)$ 및 다운링크 통신 에너지 소모량 $E_{k,n}^d(L_n^d, p_n^c)$ 은 아래와 같이 정의된다¹⁴⁾.

$$E_{k,n}^u(L_n^u, p_n^c) = \frac{1}{g_{k,n}(p_n^c)} \left(2^{\frac{L_{k,n}^u}{B\Delta}} - 1 \right) \times \left(N_0 B \Delta + \sum_{k'=1, k' \neq k}^K g_{k',n}(p_n^c) E_{k',n}^u(L_n^u, p_n^c) \right) \quad (10)$$

$$E_{k,n}^d(L_n^d, p_n^c) = \left(2^{\frac{L_{k,n}^d}{B\Delta}} - 1 \right) \times \left(\frac{N_0 B \Delta}{g_{k,n}(p_n^c)} + \sum_{k'=1, k' \neq k}^K E_{k',n}^d(L_n^d, p_n^c) \right) \quad (11)$$

수식 (10)과 수식 (11)에서 u 와 d 는 업링크와 다운링크를 각각 의미하고, $L_n^u = \{L_{k,n}^u\}_{k \in \{1, \dots, K\}}$, $L_n^d = \{L_{k,n}^d\}_{k \in \{1, \dots, K\}}$ 를 뜻한다.

3) 비행 에너지 모델: 무인 항공기의 비행을 위해 소모되는 에너지는 두 가지 모델을 고려할 수 있다¹⁷⁾. 첫 번째 모델은 오직 n 번째 프레임의 비행 속도에만 의존하는 모델 (Model 1)로 수식 (12)와 같이 정의된다.

$$E_{F,n}^c(v_n^c) = \kappa \|v_n^c\|^2 \quad (\text{Model 1}) \quad (12)$$

수식 (12)에서, $\kappa = 0.5M\Delta$ 이고, M 은 무인 항

공기의 질량을 뜻한다. 두 번째 모델은 비행 속력뿐만 아니라, 가속력의 영향을 받는 모델 (Model 2)로 아래와 같이 표현된다.

$$E_{F,n}^c(v_n^c, a_n^c) = \kappa_1 \|v_n^c\|^3 + \frac{\kappa_2}{\|v_n^c\|} \left(1 + \frac{\|a_n^c\|^2}{g^2} \right) \quad (\text{Model 2}) \quad (13)$$

수식 (13)에서, $a_n^c = \frac{v_{n+1}^c - v_n^c}{\Delta}$ 는 무인 항공기의 가속도를 나타내고, g 는 중력가속도를 뜻하며, 상수 κ_1 과 κ_2 는 [17]의 Appendix C와 동일하게 정의된다.

3.3 모바일 에너지 소모를 최소화하는 최적의 비트 할당과 오프로딩 및 항공 궤적 디자인 기술

여기서는 무인 항공기에 장착된 클라우드렛이 고려된 모바일 엣지 컴퓨팅 시스템의 전체 모바일 에너지 소모량을 최소화하는 최적의 비트 할당과 오프로딩 및 항공 궤적 디자인 기술을 연구한다. 섹션 II-2에서 다뤘던 차량 엣지 컴퓨팅 시스템은 엣지 노드인 RSU가 고정 노드였던 반면, 이번 장에서는 무인 항공기에 장착된 엣지 노드를 고려하기 때문에 최적의 비트 할당과 최적 오프로딩 디자인과 함께 무인 항공기의 최적의 궤적 디자인까지 고려해야하는 차이점이 있다. 첫 번째로, 벤치마크로 오프로딩이 고려되지 않았을 때의 경우인 로컬 컴퓨팅 시나리오에 대해 연구하고, 모바일 엣지 컴퓨팅 시스템 에너지 소모를 최소화하는 최적 기술에 대해 연구한다.

로컬 컴퓨팅 에너지 소모량: 모든 어플리케이션이 모바일 자체적으로 수행될 때의 전체 에너지를 계산한다. 테드라인 T 동안 I_k 비트 연산을 위해서, 모바일 기기 k 의 프리퀀시는 $f_k^m = C_k I_k / T$ 로 설정되어야 한다. 따라서, 로컬 컴퓨팅 시나리오의 전체 에너지 소모량은 수식 (7)과 같이 정의된다. 단, 수식 (7)에서의 v 는 모바일 기기를 나타내는 m 으로 치환하여 고려한다.

최적화 문제를 정의하기 위해, 다음의 표현을 추가적으로 정의한다. n 번째 프레임에서, 모바일 기기 k 가 업링크를 통해 클라우드렛에게 $L_{k,n}^u$ 비트를 송신하고, 클라우드렛은 $l_{k,n}^c$ 를 컴퓨팅함과 동시에 모두

계산한 $L_{k,n}^d$ 비트를 다운링크를 통해 다시 모바일 기기 k 에게 전송한다. 또한, 클라우드렛은 n 프레임에서 f_n^c 의 프리퀀시로 컴퓨팅을 수행한다. 클라우드렛은 모든 모바일 기기에게 받은 task를 수행해야하므로, 위의 정의에 따라 n 번째 프레임에서 $\sum_{k=1}^K l_{k,n}^c$ 을 계산해야 한다. 따라서, $f_n^c = \sum_{k=1}^K C_k l_{k,n}^c / \Delta$ 의 프리퀀시로 동작해야하며, 이에 따라 모바일 기기 k 을 위한 오프로딩을 수행하는데에 사용되는 클라우드렛의 컴퓨팅 에너지는 아래와 같이 계산된다.

$$E_{k,n}^c(l_n^c) = \frac{\gamma^c C_k l_{k,n}^c}{\Delta^2} \left(\sum_{k'=1}^K C_{k'} l_{k',n}^c \right)^2 \quad (14)$$

수식 (14)에서 $l_n^c = \{l_{k,n}^c\}_{k \in \{1, \dots, K\}}$ 는 n 번째 프레임에서 클라우드렛이 컴퓨팅해야 하는 모든 비트를 나타낸다. 모바일 엣지 컴퓨팅 시스템의 디자인 목표는 모든 모바일 기기의 에너지 소모량을 최소화하는 것으로서, 모바일 기기의 배치에 맞게 무인 항공기의 최적 궤도 $\{p_n^c\}$ 를 디자인하고, 최적의 업/다운링크 통신 비트 및 클라우드렛의 컴퓨팅 비트 할당 $\{L_{k,n}^u\}$, $\{l_{k,n}^c\}$, $\{L_{k,n}^d\}$ 및 오프로딩 $\{\rho_k\}$ 을 디자인하는 것으로, 이를 위해 아래와 같이 최적화 문제를 정의할 수 있다.

$$\begin{aligned} & \min_{\{L_{k,n}^u\}, \{l_{k,n}^c\}, \{L_{k,n}^d\}, \{p_n^c\}, \{\rho_k\}} \sum_{k=1}^K \sum_{n=1}^{N-2} E_{k,n}^u(L_{k,n}^u, p_n^c) \\ & \quad + \sum_{k=1}^K E_k^m((1-\rho_k)I_k) \\ \text{s.t.} \quad & \sum_{k=1}^K \sum_{n=1}^{N-2} E_{k,n+1}^c(l_{k,n+1}^c) + E_{k,n+2}^d(L_{k,n+2}^d, p_n^c) \\ & \quad + \sum_{n=1}^N E_{F,n}^c(v_n^c) \leq \epsilon, \\ & \sum_{n=1}^{N-2} L_{k,n}^u = \rho_k I_k, \\ & \sum_{n=1}^{N-2} l_{k,n+1}^c = \rho_k I_k, \\ & \sum_{n=1}^{N-2} L_{k,n+2}^d = O_k \rho_k I_k, \\ & 0 \leq \rho_k \leq 1, \\ & \sum \end{aligned} \quad (15)$$

수식 (15)에서, ϵ 는 무인 항공기의 에너지 제한 조건으로써 컴퓨팅과 비행 에너지 소모량은 ϵ 이내로 사용되어야 한다. 또한, 수식 (15)에서는 무인 항공기의 속도에만 의존하는 비행 에너지 모델 수식 (12)를 고려하였다. 이 대신, 가속력과 속력 모두에 의존하는 비행 에너지 모델 수식 (13)을 적용하는 경우, 수식 (15)에서 비행 에너지 $E_{F,n}^c(v_n^c)$ 대신 $E_{F,n}^c(v_n^c, a_n^c)$ 를 사용하고, 무인 항공기의 속도 및 가속도 관련 아래의 조건들을 추가적으로 고려한다.

$$v_{n+1}^c = v_n^c + a_n^c \Delta, \quad (16)$$

$$p_{n+1}^c = p_n^c + v_n^c \Delta + \frac{1}{2} a_n^c \Delta^2, \quad (17)$$

$$v_1^c = v_{N+1}^c = v^c, \quad (18)$$

$$\|a_n^c\| \leq a_{\max} \quad (19)$$

수식 (18)은 무인 항공기의 이착륙 속도 제한 조건을 표현한 것이고, 수식 (19)는 무인 항공기의 최대 가속력 제한 조건 a_{\max} 을 나타낸 것이다. 수식 (15)는 non-convex objective 함수와 non-convex 조건으로 구성되어 non-convex programming이다. 이에 따라, global optimal solution은 구할 수 없지만, inner convex approximation framework기반의 successive convex approximation (SCA)^[26] 방법을 적용하여 local solution을 구할 수 있다. 이 방법은 non-convex objective function과 non-convex 조건들을 적절한 convex approximation으로 대체하여 iterative solution을 산출해낸다. 각 iterative step에서의 변형된 convex problem은 generic solver (CVX MOSEK)나 Lagrangian dual variable을 사용하여 해결이 가능하다. 본 논문에서의 최적화 문제 (15)는 [17]에서와 다르게 최적의 오프로딩 비율을 추가적으로 디자인하여 추가적인 성능 이득을 볼 수 있다. 하지만, 수식 (15)는 $\{\rho_k\}$ 에 대하여 convex programming이므로 [17]에서 소개된 SCA 방법 기반의 알고리즘을 적용하면 해결이 가능하므로, 자세한 내용은 본 논문에서 생략하기로 한다.

3.4 모의 실험 및 분석

본 장에서는 제안된 무인 항공기에 장착된 클라우드렛을 고려한 모바일 엣지 컴퓨팅 시스템 최적화 기

술의 성능을 분석한다. 최적 오프로딩 디자인의 성능 이득은 섹션 II-4의 차량 엣지 컴퓨팅 시스템 실험에서 파악할 수 있으므로, 본 장에서는 complete offloading인 $\rho_k = 1$ 을 고려하여 최적 비트 할당 및 무인 항공기 최적 디자인으로 인한 성능 이득에 집중한다. 또한, 엣지노드가 고정 노드였던 차량 엣지 컴퓨팅 시스템과 다르게 이동 노드 차량이 엣지 노드이기 때문에, 추가적으로 앞서 연구한 차량의 이동 모델에 따른 최적 궤적 변화를 살펴보기로 한다. 제안 기술의 성능 비교를 통해, 세 가지 레퍼런스를 고려한다. 첫 번째로 각 프레임마다 업/다운링크 통신 비트 및 컴퓨팅 비트를 동일하게 고려하고, 이착륙 지점 사이를 동일한 속력으로 직선 비행한 경우 (no optimization), 두 번째로 동일한 속력의 무인 항공기 비행을 고려하지만, 각 프레임마다의 최적 통신 비트와 최적 컴퓨팅 비트를 할당한 경우 (optimal bit allocation), 세 번째로 각 프레임마다 통신 비트와 컴퓨팅 비트를 모두 동일하게 고려하지만, 무인 항공기의 최적 비행을 고려한 경우 (optimal trajectory)로, 수식 (15)에서 제안한 최적 항공 궤적 및 비트 할당 기술 (optimal trajectory and bit allocation)과 비교한다. 또한, 비직교 다중 접속 방식과 함께 [17]에서 소개된 직교 다중 접속 방식에서의 최적 기술과 함께 고려한다.

그림 5는 직교 다중 접속 방식을 고려하였을 때 모바일 엣지 컴퓨팅 시스템을 위해 제안된 알고리즘으로 얻어진 최적 무인 항공기의 궤적을 보여준다. $10 \times 10 m^2$ 의 사각 지역의 모서리에 별도로 표시된 $K = 3$ 개의 모바일 기기를 고려하고 (즉,

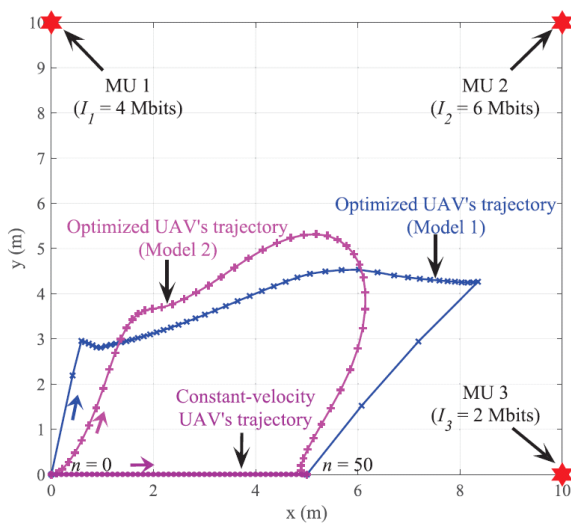


그림 5. 직교 접속 방식에서의 무인 항공기의 최적 궤적
Fig. 5. The optimized UAV's trajectory for orthogonal access

$p_1^m = (0, 10, 0)$ m, $p_2^m = (10, 10, 0)$ m, $p_3^m = (10, 0, 0)$ m), 각각의 인풋 데이터는 $(I_1, I_2, I_3) = (4, 6, 2)$ Mbits로 고려하였다. 무인 항공기의 이착륙 지점은 $p_I^c = (0, 0)$ 와 $p_F^c = (5, 0)$ 로 설정되었고, 기준 SNR $g_0/(N_0B) = -5$ dB로 설정하였다. 그림 5에서 보는 바와 같이, 본 실험에서는 섹션 III-2에서 연구한 두 가지 무인 항공기 비행 모델 Model 1과 Model 2에 대해서 살펴보았다. 두 모델 모두 Task의 양이 많은 오프로딩 기기인 모바일 기기 2에게 오래 머물도록 설계됨을 확인할 수 있다. 하지만, Model 2의 경우, 무인 항공기의 가속도에 의존하는 모델이기 때문에, Model 1에 비하여 부드럽게 궤적을 형성하는 것을 알 수 있다. 이를 통해, 시스템 설계의 실용성을 위해, 올바른 에너지 소모 모델 설계가 중요한 것을 확인할 수 있다.

그림 6은 테드라인 T 에 따른 전체 모바일 기기 에너지 소모량을 보여준다. $10 \times 10 m^2$ 의 사각 지역에 uniformly 배치된 $K = 2$ 개의 모바일 기기를 고려하고, 각각의 인풋 데이터는 $I_1 = I_2 = 8$ Mbits로 고려하였다. 무인 항공기의 이착륙 지점은 $p_I^c = (0, 0)$ 와 $p_F^c = (0, 8)$ 로 설정되었고, 기준 SNR $g_0/(N_0B) = -2.5$ dB로 설정하였으며, 무인 항공기 비행 에너지 모델은 수식 (12)를 고려하였다. 그림 6에서 보는 바와 같이, 테드라인이 짧아질수록, 모바일 기기가 모든 어플리케이션 컴퓨팅을 수행하는 로

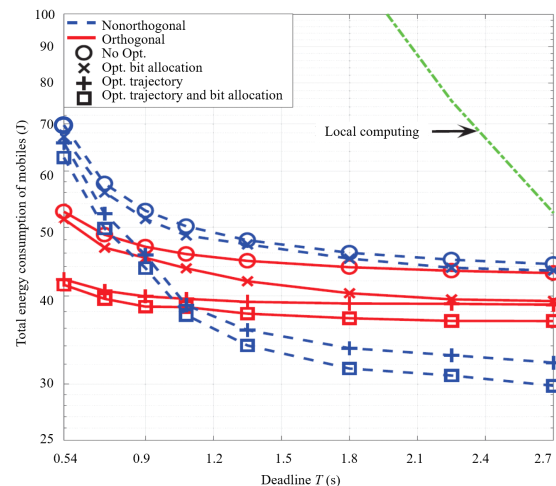


그림 6. 무인 항공기에 장착된 클라우드렛이 고려된 엣지 컴퓨팅 시스템에서 테드라인 T 에 따른 전체 모바일 기기 에너지 소모량 비교
Fig. 6. Average total energy consumption of the mobile devices as a function of the deadline T

컬 컴퓨팅에 비해 오프로딩 수행 시 굉장히 큰 에너지 효율 이득을 얻는 것을 확인할 수 있다. 또한, 무인 항공기의 궤적 디자인이 비트 할당보다 에너지 효율성 측면에서 더 큰 효과를 보임을 알 수 있다. 이에 덧붙여, 테드라인이 짧지 않은 환경에서는 비직교 다중 접속 방식이 직교 다중 접속 방식보다 성능이 더 좋다는 것이 관찰된다. 이는 더 짧은 테드라인일수록 더 많은 에너지 소모량이 발생하여 기기 간 간섭이 발생하는 비직교 다중 접속 방식이 제한되기 때문이다. 즉, 고려하는 모바일 엣지 컴퓨팅 환경의 특징에 맞게 적절한 접속 방식을 선택해야 한다.

IV. 결 론

본 논문에서는 차량이 고려되는 엣지 컴퓨팅 시스템을 크게 두 가지로 나누고, 각 시스템의 대표 시나리오에 대해 연구하였다. 자세히는, 오프로딩 기기가 차량인 차량 엣지 컴퓨팅 시스템과 오프로딩된 엣지 노드가 차량인 무인 항공기에 장착된 클라우드렛 기반의 모바일 엣지 컴퓨팅 시스템을 고려하였다. 두 시나리오 모두 한정된 배터리 기반의 오프로딩 기기와 차량을 고려한 엣지 컴퓨팅 시스템이기 때문에, 디자인 설계 방식에 있어서 공통적으로 이슈가 되는 사항을 집중하여 연구하였다. 또한, 에너지 자원이 풍부한 고정 엣지 노드를 고려하는 차량 엣지 컴퓨팅 시스템과 달리 한정된 배터리 기반의 이동 엣지 노드를 고려하는 모바일 엣지 컴퓨팅 시스템이기에, 다르게 주의하여야 할 시스템 설계 기술 요소에 대해 연구하였다. 각 시나리오에서 차량으로 인해 이슈가 되고 있는 현안들을 살펴보고, 최적화 문제를 형성하여, 솔루션을 살펴보았다. 이전 연구들과 다르게 직교 다중 접속 방식과 비직교 다중 접속 방식을 모두 고려하고, 유연한 오프로딩 비율을 고려함으로써, 다양한 서비스 분야의 미래 엣지 컴퓨팅 기반 시나리오에 적용 가능성을 높이고자 하였다.

References

[1] M. Sookhak, F. R. Yu, Y. He, H. Talebian, N. S. Safa, N. Zhao, M. K. Khan, and N. Kumar, "Fog vehicular computing: Augmentation of fog computing using vehicular cloud computing," *IEEE Veh. Tech. Mag.*, vol. 12, no. 3, pp. 55-64, Jul. 2017.

[2] W. Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob,

and A. Ahmed, "Edge computing: A survey," *Future Generation Comput. Syst.*, vol. 97, pp. 219-235, Aug. 2019.

[3] P. Pace, G. Aloï, R. Gravina, G. Caliciuri, G. Fortino, and A. Liotta, "An edge-based architecture to support efficient applications for healthcare industry 4.0," *IEEE Trans. Ind. Info.*, vol. 15, no. 1, pp. 481-489, Jan. 2019.

[4] K. Zhang, Y. Mao, S. Leng, Y. He, and Y. Zhang, "Mobile-edge computing for vehicular networks: A promising network paradigm with predictive offloading," *IEEE Veh. Tech. Mag.*, vol. 12, no. 2, pp. 36-44, Jun. 2017.

[5] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Trans. Veh. Tech.*, vol. 65, no. 6, pp. 3860-3873, Jun. 2016.

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

[7] Z. Zhou, P. Liu, Z. Chang, C. Xu, and Y. Zhang, "Energy-efficient workload offloading and power control in vehicular edge computing," in *Proc. IEEE Wirel. Commun. Netw. Conf. Wkshps.*, pp. 191-196, Barcelona, Spain, Apr. 2018.

[8] Z. Zhou, J. Feng, Z. Chang, and X. Shen, "Energy-efficient edge computing service provisioning for vehicular networks: A consensus ADMM approach," *IEEE Trans. Veh. Tech.*, vol. 68, no. 5, pp. 5087-5099, Mar. 2019.

[9] S. Li, G. Zhu, and S. Lin, "Joint radio and computation resource allocation with predictable channel in vehicular edge computing," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, pp. 3736-3741, Maui, USA, Nov. 2018.

[10] Y. Jang, J. Na, S. Jeong, and J. Kang, "Energy-efficient task offloading for vehicular edge computing: joint optimization of offloading and bit allocation," in *Proc. IEEE*

- VTC2020-Spring*, Antwerp, Belgium, May 2020.
- [11] L. Alvarez, R. Horowitz, and P. Li, "Traffic flow control in automated highway systems," *Control Eng. Practice*, vol. 7, no. 9, pp. 1071-1078, Sep. 1999.
- [12] H. Wang, X. Li, H. Ji, and H. Zhang, "Federated offloading scheme to minimize latency in MEC-enabled vehicular networks," in *Proc. IEEE GLOBECOM Wkshps.*, pp. 1-6, Abu Dhabi, UAE, Dec. 2018.
- [13] W. H. Yuan and K. Nahrstedt, "Energy-efficient soft real-time CPU scheduling for mobile multimedia systems," *ACM SIGOPS Oper. Syst. Rev.*, vol. 37, no. 5, pp. 149-163, Dec. 2003.
- [14] T. M. Cover and J. A. Thomas, *Element of Information Theory*, Hoboken, NJ, USA: Wiley, 2006.
- [15] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: Opportunities and challenges," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 36-42, May 2016.
- [16] S. Jeong, O. Simeone, A. Haimovich, and J. Kang, "Mobile cloud computing with a UAV-mounted cloudlet: Optimal bit allocation for communication and computation," *IET Commun.*, vol. 11, no. 7, pp. 969-974, May 2017.
- [17] S. Jeong, O. Simeone, and J. Kang, "Mobile edge computing via a UAV-mounted cloudlet: Optimization of bit allocation and path planning," *IEEE Trans. Veh. Tech.*, vol. 67, no. 3, pp. 2049-2063, Mar. 2018.
- [18] S. Rahman, G.-H. Kim, Y.-Z. Cho, and A. Khan, "Positioning of UAVs for throughput maximization in software-defined disaster area UAV communication networks," *J. Commun. and Netw.*, vol. 20, no. 5, pp. 452-463, Oct. 2018.
- [19] Y. Zeng, R. Zhang, and T. J. Lim, "Throughput maximization for UAV-enabled mobile relaying systems," *IEEE Trans. Commun.*, vol. 64, no. 12, pp. 4983-4996, Dec. 2016.
- [20] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, pp. 187-198, Tokyo, Japan, May 2004.
- [21] R. Shah, S. Roy, S. Jain, and W. Brunette, "Data MULEs: Modeling and analysis of a three-tier architecture for sparse sensor networks," *Ad Hoc Netw.*, vol. 1, no. 2, pp. 215-233, Sep. 2003.
- [22] P. Zhan, K. Yu, and A. L. Swindlehurst, "Wireless relay communications with unmanned aerial vehicles: Performance and optimization," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 47, no. 3, pp. 2068-2085, Jul. 2011.
- [23] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Drone small cells in the clouds: Design, deployment and performance analysis," in *Proc. IEEE Global Commun. Conf.*, pp. 1-6, San Diego, CA, USA, Dec. 2015.
- [24] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Optimal transport theory for power-efficient deployment of unmanned aerial vehicles," in *Proc. IEEE Int. Conf. Commun.*, pp. 1-6, Kuala Lumpur, Malaysia, May 2016.
- [25] C. Geng, N. Naderializadeh, A. S. Avestimehr, and S. A. Jafar, "On the optimality of treating interference as noise," *IEEE Trans. Inf. Theory*, vol. 61, no. 4, pp. 1753-1767, Apr. 2015.
- [26] G. Scutari, F. Facchinei, and L. Lampariello, "Parallel and distributed methods for constrained nonconvex optimization-Part I: Theory," *IEEE Trans. Sig. Proc.*, vol. 65, no. 8, pp. 1929-1944, Dec. 2016.

정 성 아 (Seongah Jeong)



2010년 2월 : 한국과학기술원 정
보통신공학과 졸업

2012년 2월 : 한국과학기술원 전
기 및 전자공학부 석사 졸업

2015년 8월 : 한국과학기술원 전
기 및 전자공학부 박사 졸업

2015년 9월~2016년 2월 : 한국
과학기술원 정보전자연구소 박사후 연구원

2016년 3월~2017년 8월 : Harvard John A. Paulson
School of Engineering and Applied Sciences 박사
후 연구원

2017년 9월~2019년 8월 : Samsung Research 책임 연
구원

2019년 9월~현재 : 경북대학교 전자공학부 조교수
<관심분야> 통신공학, 신호 처리

[ORCID:0000-0002-9737-0432]