

저궤도 위성-모바일 엣지 컴퓨팅 모델을 위한 동적 최적화 알고리즘

김정환, 함동호, 김태연, 곽정호
대구경북과학기술원

(ghks9876, dhham97, kim77ty, jeongho.kwak)@dgist.ac.kr

Dynamic Optimization Algorithm for LEO Satellite-Mobile Edge Computing Model

Jeonghwan Kim, Dongho Ham, Taeyeoun Kim, and Jeongho Kwak
Daegu Gyeongbuk Institute of Science and Technology

요약

본 논문에서는 일반적인 통신 시스템을 이용하기 힘든 고립된 지역에서의 위성-모바일 간 엣지 컴퓨팅 프레임워크를 제안한다. 해당 프레임워크는 한 명의 모바일 사용자와 저궤도 위성 집단 사이의 엣지 컴퓨팅 모델에서 발생하는 사용되는 파워와 지연시간 문제를 정의한다. 끊임없이 움직이는 저궤도 위성은 모바일 간의 거리가 계속 변하는 동적인 상황을 연출하므로 기존의 최적화 기법으로는 해결하기에 어려움을 겪게 된다. 따라서 본 논문은 시간에 따라 변하는 환경을 반영할 수 있는 동적 최적화 알고리즘을 제안하였다.

I. 서론

4G, 5G 모바일 셀룰러 네트워크는 빠르고 간편한 통신 서비스를 제공함에 있어서 매우 좋은 성취를 이뤄냈다. 하지만 아직 절반 이상의 인구가 지역적 특성 또는 낙후된 시설 등의 이유로 원활한 인터넷 서비스를 누리지 못하고 있다 [1]. 그래서 이를 위한 대표적인 해결책으로 최근 위성 통신 기술이 관심을 모으고 있다. 특히 저궤도 (Low Earth Orbit, LEO) 위성 기술은 낮은 비용, 지연시간으로부터 나오는 높은 효율로 인해 단연 돋보이고 있으며, 관련된 세계적인 규모의 프로젝트들 (OneWeb, SpaceX 등등...)도 활발히 진행 중이다 [2].

최근 통신 서비스는 사용자들에 의해 높은 퀄리티의 Quality of Services (QoS)를 요구받고 있다. 디바이스들의 가파른 진화는 복잡한 태스크를 수행하는 어플리케이션들의 등장을 촉진시켰고 이러한 등장은 기존보다 많은 계산을 필요로 한다. 모바일 엣지 컴퓨팅 (MEC) 은 이러한 부담을 줄여주는 대표적인 방법 중 하나이다. MEC 가 제공하는 스토리지 서비스 등은 효율적이고 빠른 데이터 처리가 가능하여 사용자 디바이스의 에너지 부담을 줄이는 QoS 제공이 가능하다 [3]. 위성 통신 분야 또한 이를 적용한 차세대 기술로써 위성 모바일 엣지 컴퓨팅 (SMEC) 기법이 논의되고 있지만 지상과 네트워크에 비하면 진척률이 훨씬 적은 게 사실이다. 기존의 MEC 연구를 그대로 적용하기에는 끊임없이 움직이는 위성의 특징상 통신 환경이 지상과 매우 달라서 추가적인 여러 고려사항이 요구된다. 따라서 본 논문에서는 SMEC 시나리오에서 위성과 모바일의 에너지와 지연시간을 최소화하는, 움직이는 위성 통신의 특징을 반영한 동적인 최적화 알고리즘을 제안한다.

II. 시스템 모델링

본 논문에서는 그림 1 과 같은 소외된 지역 (바다 위 배)에서의 사용자 한 명과 근처에 있는 LEO 위성 한 개 사이의 간단한 모델을 다루고자 한다. 자원이 풍부한

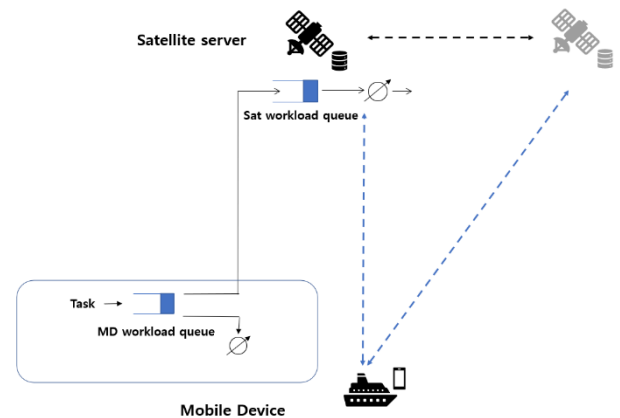


그림 1 저궤도 위성-모바일 엣지 컴퓨팅 프레임워크

도심 속이 아닌 소외된 지역 (Remote Area)은 에너지 자원이 제한적일 수 밖에 없다. 따라서 모바일에서 처리하기에 컴퓨팅 에너지가 많이 사용되는 태스크 같은 경우에 근처에 있는 LEO 위성 엣지 컴퓨팅 시스템에 오프로딩(Offloading)하여 태스크를 처리할 수 있다. 우선 모바일과 위성 엣지 서버에서의 태스크 처리 상황을 나타내기 위해 큐잉 모델 (Queueing Model)을 사용하였다. 큐는 모바일 워크로드 (workload) 큐와 위성 워크로드 큐, 두 가지가 존재하며 매 타임 슬롯마다 큐의 변화를 나타내면 다음과 같다.

$$Q_u^c(t+1) = [Q_u^c(t) - \frac{(1-\theta(t))c_u(t)}{\gamma} - \theta(t)b(t) + a(t)]^+, \quad (1)$$

$$Q_s^c(t+1) = [Q_s^c(t) - \frac{c_s(t)}{\gamma} + \theta(t)b(t)]^+, \quad (2)$$

$Q_u^c(t)$ 와 $Q_s^c(t)$ 는 각각 타임 슬롯 t 일 때, 모바일 디바이스와 위성 엣지 서버 큐의 잔여량이다. $\theta(t)$, $c_u(t)$, $c_s(t)$, γ 는 순서대로 타임 슬롯 t 일 때의 오프로딩 여부를 결정하는 오프로딩 파라미터, 모바일의 CPU 클럭 속도, 위성 엣지 서버의 CPU 클럭 속도, 처리하고 있는 어플리케이션 데이터 처리 밀도 (cycles/bits) 이다. $\theta(t)$ 는 오프로딩 할 땐 1, 오프로딩 하지 않을 땐 0 의

값을 가진다. $a(t), b(t)$ 는 타임 슬롯 t 일 때, 각각 모바일 워크로드 큐에 들어오고 나가는 데이터 양이다. 본 논문의 목적이 위성과 모바일의 전체적인 파워 및 지연시간의 최소화인 만큼 파워 모델과 지연시간 모델을 다음과 같이 정의하도록 한다.

$$P_u^c(c_u(t)) = \alpha c_u(t)^3 + \beta \quad (3)$$

$$P_s^c(c_s(t)) = \alpha' c_s(t)^3 + \beta' \quad (4)$$

$$P_{sys}(t) = (P_u^c(t) + \theta(t)p_u^T(t) + wP_s^c(t)) \quad (5)$$

$P_u^c(c_u(t)), P_s^c(c_s(t))$ 는 각각 타임 슬롯 t 에서의 모바일의 소모 파워, 위성 엣지 서버의 소모 파워이다.

$P_{sys}(t), p_u^T(t)$ 는 타임 슬롯 t 일 때, 정의한 시스템 파워, 모바일의 전송 파워이며, $\alpha, \alpha', \beta, \beta', w$ 는 모두 값을 조정해주는 상수 값이다. 일반적으로 사용되는 CPU 클럭 속도에 따른 파워 모델을 사용하였다.

$$l^p(t) = \frac{2d(t)\theta(t)}{c} \quad (6)$$

$$L_{sys}(t) = l^p(t) \quad (7)$$

지연시간 모델은 다음과 같다. $l^p(t)$ 는 타임 슬롯 t 에서의 전파 지연시간, $d(t)$ 는 타임 슬롯 t 에서의 위성과 모바일 사이 거리, $L_{sys}(t)$ 는 정의한 시스템 지연시간이다. 일반적으로 적용되는 네트워크 지연시간 중 전파 지연시간만 적용하였는데, 나머지 지연시간들은 위성과의 거리가 커서 전파 지연시간에 비해 무시된다고 가정하였다. 위 수식들을 조합하여 우리의 목적함수는 다음과 같이 정의했다.

$$(p_u^T, \theta, c_u, c_s) \min (J \cdot \overline{L_{sys}} + \overline{P_{sys}}) \quad (8)$$

$\overline{L_{sys}}, \overline{P_{sys}}, J$ 는 전체 시간 동안의 시간평균 시스템 지연시간, 시간평균 시스템 파워, 이 둘 간의 상관계수이다.

III. 제안된 알고리즘

본 연구는 한 타임 슬롯마다 동적으로 변화하는 위 목적함수를 풀기 위해 Lyapunov 최적화 기법을 제안한다. 제안된 알고리즘 모식도는 그림 2에 나타나 있다. 주요 원리는 다음과 같다. 매 타임 슬롯 t 에서 모바일 워크로드 큐에 처리해야 하는 태스크가 들어오게 된다. 그림 2 이 태스크가 처리될 장소를 모바일과 위성 엣지 서버 둘 중 골라야 하는데, 알고리즘이 둘 중 어느 것이 더 효율적인지 계산하여 오프로딩 여부를 결정하게 된다. 만약 모바일에서 처리하게 되면 모바일 자체 자원을 사용하게 되고, 위성 엣지 서버로의 오프로딩이 결정되면 위성의 자원을 사용하게 된다. 이후 다음 타임슬롯을 위해 각각의 워크로드 큐를 업데이트하면 한 타임 슬롯의 알고리즘 작동이 끝이 나게 되며, 이 과정을 매 타임슬롯마다 반복하여 진행한다.

IV. 결론

본 논문에서는 기존의 위성통신에서 더 나아가 모바일과 위성의 컴퓨팅 자원을 효율적으로 활용할 수 있는 간단한 저궤도 위성-모바일 엣지 컴퓨팅 프레임워크를 제시하였다. 그리고 일반적인 위성 엣지 컴퓨팅 연구에서 미흡하게 고려하는 위성의 움직임 특성을 반영한 시스템 모델을 제시하였다. 시간에 따라 동적이며

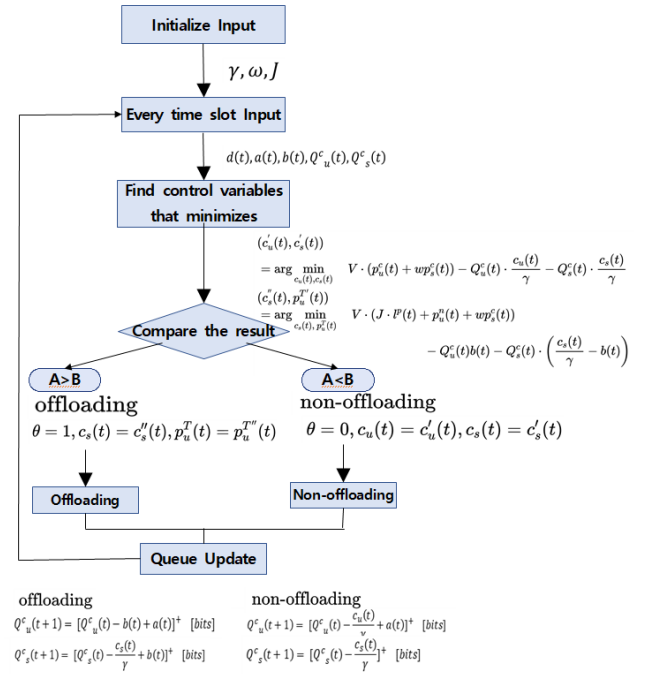


그림 2 동적 최적화 알고리즘

시스템의 소모 파워와 지연시간을 최소화하는 목적함수를 풀고자 Lyapunov 최적화 기법에 착안한 동적 최적화 알고리즘을 제안하였다. 향후 연구는 조금 더 현실적인 상황을 나타내기 위해 시스템 모델을 정밀하게 조정하고 하나의 위성과 모바일 디바이스가 아닌, 여러 위성-여러 모바일 디바이스에서의 연구를 진행하고자 한다.

ACKNOWLEDGMENT

이 논문은 2022년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.2022-0-00704, 초고속 이동체 지원을 위한 3D-NET 핵심 기술 개발)

참 고 문 헌

- [1] T. Azzarelli, "Onweb global access," in *Proc. Global Conf. Space Inf. Soc. (GLIS)*, Jun. 2016, pp. 1-24.
- [2] V. L. Foreman, A. Siddiqi, and O. D. Weck, "Large Satellite Constellation Orbital Debris Impacts: Case Studies of OneWeb and SpaceX Proposals," in *Proc. AIAA Space and Astronautics Forum and Expo*, 2017.
- [3] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE communications surveys & tutorials*, vol. 19, no. 4, pp. 2322-2358, 4th Quart. 2017.