

# 저궤도 위성 엣지 컴퓨팅 시스템을 위한 예측 정보 기반 연산 오프로딩 기법

남관교, 이시현\*

한국과학기술원 전기및전자공학부

gwankyo@kaist.ac.kr, \*sihyeon@kaist.ac.kr

## Prediction-based Computation Offloading Strategy for LEO Satellites Edge Computing

Nam Gwam Kyo, Lee Si-Hyeon\*

한국과학기술원 전기및전자공학부

### 요약

본 논문은 지상의 사용자가 저궤도 위성에 연산 작업을 오프로딩하는 상황을 고려하여, 가상 머신의 이동 에너지 제약 아래 연산 작업의 전송 에너지와 연산 에너지의 합을 최소화하기 위한 오프로딩 최적화 문제를 다룬다. Lyapunov 최적화 기법을 도입하여 최적화 문제를 재구성하고, 저궤도 위성의 위치예측을 활용하여 결정을 내리는 예측 정보 기반 알고리즘을 제안하여 예측 정보를 활용하지 않았을 때 대비 성능 향상을 입증한다.

### I. 서론

차세대 통신에서 커버리지 향상 및 전송률 증대를 위해 저궤도 위성 활용이 큰 주목을 받고 있다. StarLink, OneWeb 등과 같은 기업들의 투자 및 개발로 저궤도 위성 통신의 상용화가 가속화되고 있다. 이에 따라 저궤도 위성에 엣지 서버를 탑재하여 연산 작업을 수행하는 저궤도 위성 엣지 컴퓨팅 시스템에 대한 연구가 활발히 진행되고 있다 [1].

본 논문에서는 지상의 사용자가 저궤도 위성에 연산 작업을 오프로딩 하는 상황에서 종합 에너지 소모량을 가상 머신 이동 에너지의 제약 아래 최소화하기 위한 오프로딩 결정 최적화를 다룬다. 저궤도 위성은 정해진 궤도를 따라 이동한다는 점을 기반으로, 저궤도 위성 위치 정보를 바탕으로 미래 정보를 예측하여 활용하는 예측 정보 기반 Lyapunov 알고리즘을 제안하고자 한다.

### II. 본론

#### 1. 시스템 모델

본 논문에서는 그림 1과 같이 지상의 사용자와 여러 궤도를 도는 저궤도 위성  $M$ 개가 있으며, 각 저궤도 위성에는 연산 처리가 가능한 엣지 서버가 탑재되어 있음을 가정한다. 지상의 사용자는 시간 슬롯  $t \in \{0, 1, \dots, T-1\}$ 마다 생성되는 연산 작업을 하나의 저궤도 위성을 선택하여 오프로딩하며, 선택된 저궤도 위성은 탑재된 엣지 서버를 통해 연산 작업을 처리한 후 결과를 사용자에게 전달한다. 오프로딩 결정 변수는  $x_i(t)$ 로 정의한다. 만약 사용자가 슬롯  $t$ 에서  $i$ 번째 저궤도 위성에게 연산 작업을 오프로딩하면  $x_i(t) = 1$ , 이외에는  $x_j(t) = 0$ 이다. 오프로딩 결정은 제어 센터에서 내리며, 제어 센터는 전체 상황을 인지하고 있다고 가정한다 [2].

연산 작업을 처리하는 과정에는 전송 에너지, 연산 에너지가 소모된다. 사용자가 저궤도 위성에 연산 작업을 전송할 때 소모되는 에너지는

$$E_i^{trans}(t) = S(t) / W \log(1 + \frac{P_{trans} h_i(t)}{\sigma^2})$$

$P_{trans}, h_i(t)$ 는 각각 연산 작업의 크기, 대역폭, 잡음 전력, 전송 파워, 채널 이득이다. 본 논문은 채널 이득으로 기본 경로 손실과 위성 안테나 이득을 고려하였으며[3], 이는 지상의 사용자와 저궤도 위상 간의 거리 및 고도각에 의해 결정된다. 고도각 90도 근처의 저궤도 위성이 일반적으로

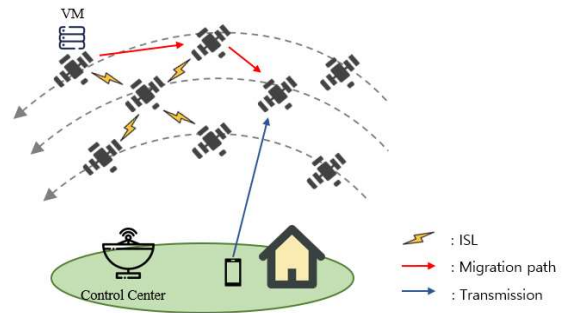


그림 1. 시스템 모델

높은 채널 이득을 가진다. 저궤도 위성이 연산 작업을 처리할 때 소모되는 연산 에너지는  $E_i^{comp}(t) = \epsilon (f_i(t))^2 K(t)$ 으로 표현되며,  $\epsilon, f_i(t), K(t)$ 는 각각 CPU cycle 당 예상 에너지 소모량, 저궤도 위성의 연산 능력, 연산 작업의 CPU cycle 요구량이다. 각 저궤도 위성은 고정된 기본 연산 능력을 가지며 시간에 따라 일정 변동량이 있음을 가정한다. 시간 슬롯  $t$ 에서 종합 에너지 소모량은  $E^{tot}(t) = \sum_{i=1}^M x_i(t) (E_i^{trans}(t) + E_i^{comp}(t))$ 로

표현된다. 연산 작업 결과는 크기가 작으므로 다운로드 상황은 무시한다.

본 논문에서는 저궤도 위성이 연산 작업을 수행하기 위해서 추가 정보가 필요함을 가정한다. 따라서 추가 정보를 가상 머신(VM)에 담아 연산 작업을 수행하는 저궤도 위성에 전송한다. 저궤도 위성은 가장 근접한 상, 하, 좌, 우 4개의 저궤도 위성과 항상 위성 간 연결(ISL)이 가능하다고 가정하며, VM은 그림 1과 같이 멀티 홉 방식으로 이동한다. ISL 채널은 자유공간 경로 손실만을 고려하며, VM의 이동 최단 경로는 Dijkstra 알고리즘을 이용해 찾는다. VM의 이동에 소모되는 에너지를 이동 에너지  $E^{mig}(t)$ 로 정의하였다. 자원이 부족한 저궤도 위성 네트워크의 특성상 잦은 VM 이동은 네트워크에 손해를 주므로 이동 에너지 제약을 둔다.

#### 2. 최적화 문제 및 제안 알고리즘

본 논문에서는 종합 에너지 소모량을 이동 에너지의 제약 아래에 최소화하기 위해 연산 오프로딩 결정 최적화를 목표로 한다. 저궤도 위성의 빠른 움직임으로 인해 네트워크가 시간에 따라 역동적으로 변하기 때문에 장기적인 관점에서 최적화를 진행한다. 최적화 문제는 다음과 같다.

$$\begin{aligned}
& \text{Minimize} && \lim_{t \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} E^{tot}(t) \\
& \text{Subject to} && \lim_{t \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} E^{mig}(t) \leq C \quad (1)
\end{aligned}$$

목적 함수인 long-term 평균 종합 에너지 소모량을 최소화하기 위해서는 매 슬롯마다 고도각 90도 근처의 연산 능력이 높은 저궤도 위성을 선택하는 것이 이상적이다. 하지만 저궤도 위성의 빠른 움직임으로 인해 고도각과 연산 능력은 매 슬롯마다 변하며, 항상 고도각 90도 근처의 저궤도 위성을 선택하려면 VM을 계속 이동시켜야 한다. 이는 높은 이동 에너지를 초래하여 제약 조건을 지키기 어렵게 만든다. 즉, 목적 함수 최소화와 제약 조건의 보장 사이에 trade-off가 존재하며, 역동적인 네트워크에서 목적 함수와 제약 조건 사이에 trade-off가 존재할 경우 Lyapunov 최적화 알고리즘이 유용하게 사용된다.

Lyapunov 알고리즘에서는 가상 큐를 도입하여 목적 함수와 제약 조건의 균형을 조절한다. 큐의 안정성을 보장하는 것을 목표로 하며 큐의 안정성 ( $Q(T-1) < \infty$ )은 제약 조건 보장 및 역동적인 네트워크에서 시스템의 안정성을 의미한다[4]. 큐는 다음과 같이 업데이트된다.

$$Q(t+1) = \max[Q(t) + E^{mig}(t) - C, 0] \quad (2)$$

저궤도 위성은 정해진 궤도를 따라 움직이므로 특정 시점에 저궤도 위성의 위치를 정확하게 알 수 있다고 가정한다. 저궤도 위성의 위치 정보를 통해 전송 에너지와 연산 에너지의 평균값을 예측할 수 있다. 따라서  $K$  슬롯 동안의 정보를 예측하여 이용하는 예측 정보 기반 Lyapunov 알고리즘을 구현할 수 있고, 문제(1)는 Lyapunov drift-plus penalty[4]에 의해 다음과 같이 변환된다.

$$\begin{aligned}
& \text{Minimize} && \sum_{t=nK}^{nK+K-1} VE^{tot}(t) + Q(nK)E^{mig}(t) \quad (3) \\
& x(nK), \dots, x(nK+K-1)
\end{aligned}$$

$V$ 는 목적 함수와 큐의 안정성 간의 trade-off 상수이다. 각 프레임  $n \in \{0, 1, 2, \dots, N-1\}$ 은  $K$  슬롯으로 이루어져 있고, 프레임의 첫 번째 슬롯에서 나머지  $K-1$  슬롯의 정보를 예측하여  $K$  슬롯 동안의 오프로딩 결정을 내린다. 이후 큐를 (2)와 같이 업데이트한다. 문제(3)는 최단 경로 찾기 문제와 동일하여 Dijkstra 알고리즘을 사용해 해를 구할 수 있다.

### 3. 결과

본 논문에서 제안한 알고리즘의 이론적 성능은 다음과 같다.

$$\begin{aligned}
& \lim_{NK \rightarrow \infty} \frac{1}{NK} \sum_{t=0}^{NK-1} E\{E^{tot}(t)\} \leq E^{tot*} + \frac{B}{V} \\
& \lim_{NK \rightarrow \infty} \frac{1}{NK} \sum_{t=0}^{NK-1} E\{Q(t)\} \leq \frac{B + VE_{max}^{tot}}{\eta}
\end{aligned}$$

$V$ 에 따라 목적 함수와 큐 길이가  $O(1/V, V)$  trade-off를 가짐을 확인할 수 있다.

시뮬레이션은 그림 1과 같은 환경을 Matlab을 통해 구현해 진행하였고, 현재의 정보만을 이용하는 1 slot 알고리즘과 예측 정보 기반 알고리즘인 3, 5, 7 slot 예측 알고리즘을 비교하였다. 그림 2에서 목적 함수와 평균 큐 길이 모두 시간에 따라 수렴함을 확인하였고, 평균 큐 길이의 수렴은 큐와 시스템의 안정성을 의미한다. 예측 정보를 이용하였을 때 목적 함수와 평균 큐 길이 모두 성능이 향상하였으며, 이는 예측 정보를 활용하면 VM의 이동 경로를 더 효율적으로 계획할 수 있기 때문이다. 즉, 90도 근처의 높은 연산 능력의 저궤도 위성으로 적은 이동 에너지를 소모하며 이동할 수 있어 종합 에너지 소모량과 큐 길이 모두 감소한다. 가장 많은 슬롯을 예측한 7 slot 알고리즘은 평균 큐 길이는 가장 짧지만, 예측 알고리즘 중 목적 함수는 가장 크다. 이는 많은 슬롯을 예측할수록 효율적인 VM

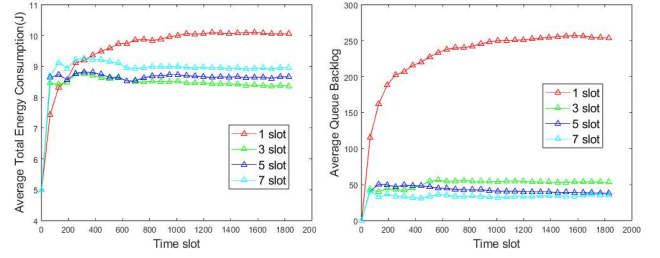


그림 2. 시간에 따른 (a) 평균 종합 에너지 소모량 (b) 평균 큐 길이

동선을 계획하지만, 예측 오차가 누적되어 목적 함수에서 손해를 보기 때문이다.

그림 3은 목적 함수와 평균 큐 길이 간의 trade-off를 보여준다.  $V$ 가 증가함에 따라 큐 안정성보다 목적 함수 최소화의 중요도가 증가해 평균 큐 길이는 증가한다. 따라서 상황에 따라 적절한  $V$  값을 선택하여 목적 함수와 큐 안정성 성능을 조절할 수 있다.

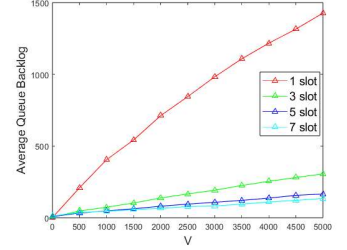


그림 3.  $V$ 에 따른 평균 큐 길이

### III. 결론

본 논문에서는 지상의 사용자가 저궤도 위성에 연산 작업을 오프로딩하는 상황에서 종합 에너지 소모량을 이동 에너지의 제약 아래 최소화하기 위한 오프로딩 결정 최적화 문제를 제안하였다. 최적화 문제를 Lyapunov 최적화 알고리즘 구조 아래에서 가상 큐를 도입한 뒤 문제를 재구성하였다. 저궤도 위성이 정해진 궤도를 따라 이동한다는 특징을 이용해 미래 시간 슬롯의 정보를 예측하여 사용하는 예측 정보 기반 알고리즘을 제안하였으며, 시뮬레이션을 통해 현재 정보 기반 알고리즘에 비해 예측 정보 기반 알고리즘의 성능 향상을 확인하였다.

### ACKNOWLEDGMENT

본 연구는 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.RS-2024-00360387, No.RS-2024-00398948).

### 참고 문헌

- [1] Xie, Renchao, et al. "Satellite-terrestrial integrated edge computing networks: Architecture, challenges, and open issues." *Ieee Network* 34.3 (2020): 224-231.
- [2] C. Li, Y. Zhang, R. Xie, X. Hao and T. Huang, "Integrating Edge Computing into Low Earth Orbit Satellite Networks: Architecture and Prototype," in *IEEE Access*, vol. 9, pp. 39126-39137, 2021, doi: 10.1109/ACCESS.2021.3064397.
- [3] 3rd Generation Partnership Project (3GPP), "Solutions for NR to support Non-Terrestrial Networks (NTN)," Technical Report (TR) 38.821, Version 16.2.0, April 2023.
- [4] Neely, Michael. *Stochastic network optimization with application to communication and queueing systems*. Springer Nature, 2022.