

LEO 클러스터의 실시간 하드웨어 상태를 반영한 태스크 스케줄링 및 자원 최적화

김정환, kwakjeongho
대구경북과학기술원

ghks9876@dgist.ac.kr, jeongho.kwak@dgist.ac.kr

Task Scheduling and Resource Allocation Reflecting Real-Time Status in LEO Cluster

Kim Jeonghwan, Kwak Jeongho
DGIST

요약

저궤도 (LEO) 위성 네트워크를 활용한 통신 및 태스크 처리가 각광받고 있는 요즘, 이를 위한 최적화 연구가 많이 진행되고 있다. 하지만 지상의 최적화 방식을 위성 플랫폼에 그대로 적용하다 보니, 저궤도 위성 특성을 제대로 반영하지 못해 현실적이지 못한 부분들이 많이 발생하고 있다. 따라서 본 논문에서는 저궤도 위성 클러스터에서의 현실적인 태스크 스케줄링 및 자원 분배 최적화를 위해, 위성의 하드웨어 특성을 동적으로 반영하는 아키텍처 및 알고리즘을 제안한다.

I. Introduction

6G 시대로 전환됨에 따라, 다양한 환경에서 안정적으로 동작하는 인터넷 서비스의 중요성이 부각되고 있다. 이를 위해 빠르게 발전 중인 저궤도 (LEO) 위성 네트워크가 핵심 기술로 주목받고 있다. 특히 최근에는 단순한 통신 중계 역할을 넘어 위성 네트워크의 온보드 컴퓨팅을 활용한 태스크 처리 및 스케줄링에 대한 연구가 활발히 진행되고 있으나, 위성 하드웨어의 제한과 우주의 제약적인 환경으로 인해 여전히 많은 도전 과제가 존재하는 현실이다. 예를 들어 소형화된 LEO 위성은 배터리 용량이 매우 제한적인데, 하드웨어 특성상 배터리 잔량이 부족해질수록 위성의 수명이 급격하게 감소한다는 문제가 있다. 또한 우주에서는 열 배출이 힘들어 위성의 하드웨어 온도 조절의 중요성이 더욱 강조된다. 우주 환경에서의 위성 수리가 거의 불가능하다는 점을 고려할 때 이런 문제들은 매우 치명적이다. 이에 본 논문에서는 위성 배터리의 실시간 충전 특성과 연산량에 따른 온도 변화를 모델링하고, 이를 동적으로 고려한 태스크 스케줄링 및 자원 분배 최적화 기법을 제안한다.

II. System Architecture

A. Task Model

우리는 시간 슬롯이 $t = \{0, 1, 2, \dots\}$ 로 이루어진 시스템을 고려하여, 각 시간 슬롯의 길이는 Δt 이다. 매 시간 슬롯 t 에서 각각의 위성 k 는 비트 단위의 태스크 $a_k(t)$ 를 생성하는데, 이는 모든 시간 슬롯에서 i.i.d. 조건 (independent and identically distributed) 을 만족한다고 가정한다. 또한 $a_k(t)$ 는 a_{max} 이하로 제한되며, $\mathbb{E}[a_k(t)] = \lambda$ 를 만족한다.

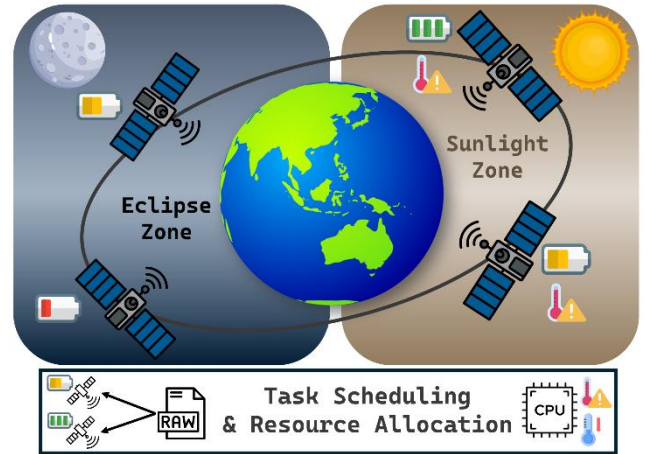


Figure 1. LEO Cluster System Architecture

B. Processing and Networking Model

매 시간 슬롯 LEO 위성 k 의 온보드 컴퓨팅 CPU 클럭 주파수는 Δt 시간동안의 사이클 수 기준으로 $s_k(t) = \{s_1, s_2, \dots, s_{max}\}$ 와 같이 가정된다. 생성된 태스크 1 bit 를 처리하는데 필요한 사이클 수는 처리밀도 γ 로 고려되며, 시간슬롯 t 일 때 LEO 위성 k 에서 같은 궤도의 LEO 위성 m 으로의 오프로딩 결정은 $\theta_{k,m}(t) = \{0, 1\}$ 로 표현된다. 위성의 프로세싱 전력 소모는 CPU 클럭 주파수에 따라 $P_k(s_k(t)) = \alpha s_k(t)^3 + \beta$ 로 모델링된다. 이에 따라 LEO 위성의 시간 슬롯에 따른 작업 대기열과 배터리 모델을 다음과 같이 표현할 수 있다.

$$Q_k(t+1) = \left[Q_k(t) - \frac{(1 - \sum_m^M \theta_{k,m}(t))s_k(t)}{\gamma} - \sum_m^M \theta_{k,m}(t)r(t) + a_k(t) + \sum_m^M \theta_{m,k}(t)r(t) \right]^+, \quad (1)$$

$$B_k(t+1) = \left[B_k(t) - P_k(s_k(t)) - \sum_m^M \theta_{k,m}(t)P_{net} + P_{solar,k}(t) \right]^+ \quad (2).$$

$r(t)$ 는 시간 슬롯 t 일 때 ISL (inter-satellite link)을 통해 전송 가능한 데이터 용량이고, P_{net} 은 오프로딩 시 소모 전력, 그리고 $P_{solar,k}(t)$ 는 시간 슬롯 t 일 때 태양광 패널을 통해 얻게되는 전력량이다.

III. Algorithm

본 연구에서는 제한된 시간 내에 태스크를 처리하여 서비스를 제공하고, 위성의 평균 배터리 상태를 최소화하는 것을 목표로한다. 이를 위해 각 위성들의 매 타임 슬롯의 작업 대기열과 배터리의 충전 상태 및 위성 탑재체의 온도를 고려하여 태스크 스케줄링 및 자원 최적화를 최적화한다. 우리의 long-term problem은 다음과 같이 설계된다.

$$\begin{aligned}
(\mathbf{P1}): \min_{s, \theta} & \left(\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_k \{B_{max} - B_k(t)\} \right), & (3) \\
s.t. (C1): & \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_k Q_k(t) < \infty, \\
s.t. (C2): & \sum_m \theta_{k,m}(t) \leq 1, \quad \forall k, \forall m, \forall t, \\
s.t. (C3): & Temp_k(t) < Temp_{max}.
\end{aligned}$$

$(s, \theta) \triangleq (s_k(t), \theta_{k,m}(t) : k \in \mathcal{K}, m \in \mathcal{M}, t \in \{0, 1, 2, \dots, \infty\})$ 와 같이 정의된다. (C1)은 각 위성의 작업 대기열이 유한하여 서비스 처리 시간 또한 유한하다는 의미이며 (C2)는 한 시간 슬롯에서 위성은 하나의 다른 위성에게만 오프로딩 가능하다는 의미이다. 또한 (C3)는 각 위성 탑재체의 온도는 제약온도를 넘어설 수 없다는 의미이다.

이 문제를 풀기 위해, 우리는 Lyapunov 최적화 기법을 활용하여 원래의 long-term problem을 slot-by-slot problem으로 변환한다. 먼저 Lyapunov drift를 다음과 같이 정의한다.

$$\Delta(L(Q(t))) = \mathbb{E}\{L(Q(t+1)) - L(Q(t)) | Q(t)\}, \quad (4)$$

$L(Q(t)) = \frac{1}{2} \sum_k Q_k(t)^2$ 이다. 배터리 상태를 반영하기 위해, Lyapunov drift에 penalty 항을 더한 Lyapunov drift-plus-penalty 함수를 다음과 같이 정의한다.

$$\Delta(L(Q(t))) + V \mathbb{E}\{B_{max} - B_k(t) | Q(t)\}. \quad (5)$$

V 는 배터리 상태와 작업 대기열 안정성 간의 트레이드오프를 조절하는 파라미터이다. 이후 [1]과 유사한 방식으로 수식 (5)의 상한을 최소화함으로써, 배터리 상태와 작업 대기열 안정성을 동시에 최적화할 수 있다. 알고리즘 1을 간단히 표현하자면, 매 시간 슬롯마다 수식 (5)의 상한을 계산하여 최소값을 도출하는 스케줄링 및 자원 분배 케이스를 선별한다.

Algorithm 1 Joint Task Scheduling and Resource Allocation

```

1: for each time slot  $t$  do
2:   for each LEO satellite  $k$  do
3:      $A \leftarrow \min_{s_k(t), \theta_{m,k}(t)} \left( V(P_k(s_k(t)) - P_{solar,k}(t)) \right.$ 
4:        $\left. - \left( \frac{s_k(t)}{\gamma} - a_i(t) - \sum_m \theta_{m,k}(t)r(t) \right) Q_i(t) \right)$ 
5:      $B \leftarrow \min_{s_k(t), \theta_{m,k}(t), \theta_{k,m}(t)} \left( V(P_{net} - P_{solar,k}(t)) \right.$ 
6:        $\left. - \left( \sum_m \theta_{k,m}(t)r(t) - a_i(t) \right.$ 
7:        $\left. - \sum_m \theta_{m,k}(t)r(t) \right) Q_i(t) \right.$ 
8:        $\left. + \sum_m \theta_{m,k}(t)r(t) Q_m(t) \right)$ 
9:   if  $A = \min(A, B)$  then
10:    Do local computing
11:   else if  $B = \min(A, B)$  then
12:    Offload to other LEO satellite
13:   end if
14: end for
15: end for

```

IV. Evaluation

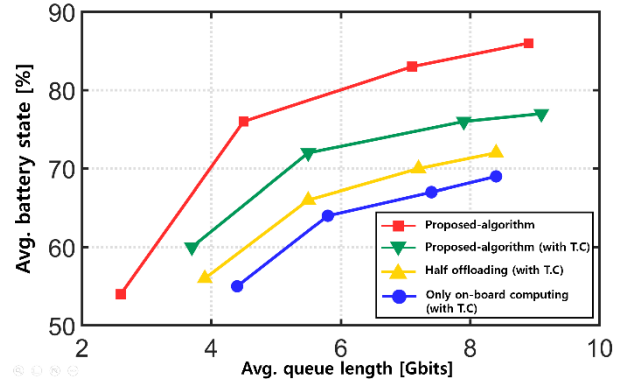


Figure 2. Battery State – Queue Length Trade-Off

Proposed-algorithm은 앞서 제한한 알고리즘이되, LEO 위성 탑재체 온도 제약조건이 고려되지 않는다. T.C (Temperature Constraint)는 이를 고려한다는 의미이며, Half offloading은 각 LEO 위성의 작업 대기열 길이가 일정 이상이면 다른 위성으로 오프로딩을 하고, 이하이면 온보드 컴퓨팅으로 처리함을 의미한다. 또한 Only on-board computing은 오프로딩하지 않고 온보드 컴퓨팅으로만 작업 대기열을 처리함을 나타낸다. 이를 통해 제안한 알고리즘이 위성의 물리적인 상태를 반영하고도 실시간 최적화를 잘 진행하여 다른 비교 알고리즘들에 비해 낮은 평균 작업 대기열, 높은 평균 배터리 상태, 즉 좋은 성능을 도출한다는 것을 Figure 2를 통해 알 수 있다.

ACKNOWLEDGMENT

이 논문은 2025년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.RS-2022-II220704, 초고속 이동체 지원을 위한 3D-NET 핵심 기술 개발)

참고 문헌

- [1] M. Neely, "Stochastic network optimization with application to communication and queueing systems," Synthesis Lectures on Communication Networks, pp. 1-211, 2010.