KICS The Korean Institute of Communications and Information Sciences

Available online at www.sciencedirect.com
ScienceDirect

ICT Express 10 (2024) 1212–1219

www.elsevier.com/locate/icte

DCOOL: Dynamic computation offloading and resource allocation for LEO satellite-assisted edge computing in a ground-space integrated framework

Jeonghwan Kim, Jeongho Kwak*

Electrical Engineering and Computer Science (EECS), Daegu Gyeongbuk Institute of Science and Technology (DGIST), Daegu, Republic of Korea

Received 25 February 2024; received in revised form 3 July 2024; accepted 23 September 2024
Available online 26 September 2024

Abstract

Despite the rapid growth of the Internet industry, the provision of full Internet service to remote regions is still challenging. As a solution, the combination of Low Earth Orbit (LEO) satellite communication and Mobile Edge Computing (MEC) is gaining attention. However, considering the high speed of LEO satellites in network environments remains a significant challenge. To this end, this paper introduces a dynamic computation offloading and resource allocation framework in the LEO satellite MEC architecture. Using Lyapunov optimization, we propose an efficient *DCOOL* algorithm to minimize average power consumption and propagation delay constrained by queue stability. Finally, comparative analysis and simulations demonstrate the superior performance of *DCOOL* while achieving lower power consumption and stable workload processing.

Keywords: LEO satellite communication; Lyapunov optimization; Mobile edge computing; Satellite edge computing

1. Introduction

Despite significant growth in the internet industry, providing full internet service to remote and underserved regions remains challenging. Satellite communication has garnered interest for improving terrestrial communication through wide coverage, abundant spectrum, and minimal interference [1]. Notably, the close distance of approximately 500 km between terrestrial stations and Low Earth Orbit (LEO) satellites has significantly reduced the propagation delay to 1–4 ms. Therefore, researchers have extensively explored LEO satellite networks, addressing ongoing challenges in a comprehensive review [2,3]. For instance, they have attempted to incorporate Software Defined Networking (SDN) and Network Function Virtualization (NFV) to advance the LEO satellite network [4,5]. Additionally, recent efforts have focused on enhancing performance by optimizing routing with the latest digital twin techniques in satellite network environments [6,7].

The rise of smartphone computation-demanding applications has increased the need for intensive data processing. Although cloud computing and mobile edge computing (MEC) in terrestrial networks address this with rich computing resources and code offloading techniques, processing in LEO satellites still faces limitations due to off-grid battery supply and their extremely high mobility. For example, researchers have introduced various resource allocation techniques in integrated MEC and STN environments, known as Satellite Mobile Edge Computing (SMEC) [8–10]. Additionally, a study has been presented where the 3-tier computing architecture of Local-MEC-CLOUD collaborates to optimize SMEC [11]. They proposed CPU clock frequency and/or transmit power control algorithms in static environments, such as static wireless channels and static workload arrivals, aiming to minimize the energy consumption of mobile devices or LEO satellites. In fact, since the space environments are considerably different from that in terrestrial networks, we must consider the unique characteristics of LEO satellites, such as their rapid mobility, constrained computing resources, and highly dynamic channel conditions between terrestrial stations and LEO satellites [12]. However, the existing SMEC studies [8–11] did not totally consider every characteristic of the dynamic space environments. Therefore, it is essential to consider appropriate resource allocation and code offloading decision techniques for a practical and dynamic SMEC environment.

The key contributions of this work are as follows: (1) We formulate a problem to minimize the average weighted

* Corresponding author.
E-mail addresses: ghks9876@dgist.ac.kr (J. Kim), jeongho.kwak@dgist.ac.kr (J. Kwak).

Peer review under responsibility of The Korean Institute of Communications and Information Sciences (KICS).

https://doi.org/10.1016/j.icte.2024.09.014

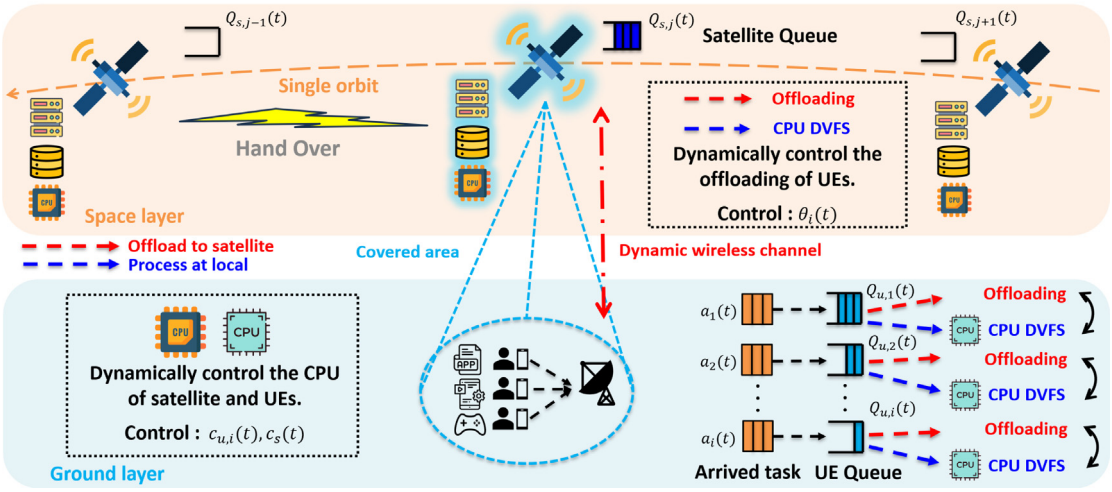


Fig. 1. Dynamic optimization system for LEO satellite-assisted edge computing on ground-space integrated framework.

sum of power consumption and propagation delay, constrained by processing delay, for the dynamic SMEC framework. (2) Using Lyapunov optimization theory [13], we develop a low-complexity dynamic optimization algorithm called *DCOOL*, which optimizes SMEC offloading decisions and CPU clock frequency scaling by removing unnecessary procedures. (3) Through extensive simulations, we evaluate the proposed *DCOOL* algorithm in terms of propagation delay, processing queue backlog, and power efficiency compared to existing algorithms.

2. System model

2.1. Framework model

Fig. 1 illustrates a ground-space integrated MEC network designed to serve the computational demands of user equipments (UEs). On the ground layer, there are I UEs where the set of UEs are represented by $\mathcal{I} = \{1, \ldots, I\}$. On the space layer, multiple LEO satellites share the same orbit as they pass over the target area. To coordinate code offloading to these LEO satellites and perform centralized control, we consider a gateway (GW) with a centralized SDN controller located near the user equipment (UEs) on the ground layer [14]. The GW has access to essential system information, such as the channel state information (CSI) of LEOs, for analysis. The GW can establish a feeder link with a single visible satellite during a specific time period. The feeder link between the visible satellite and the GW dynamically changes as satellites move across the sky. When the connected single satellite moves away from the target area, it hands over its workloads to the closest next satellite in the same orbit. We assume that the links between LEO satellites use optical communication instead of RF for ease of analysis. This allows for the rapid transfer of workload to the next satellite.

Since managing system performance continuously is challenging, we model it as a time-slotted system, with each time slot denoted by $t \in \mathcal{T} = \{0, 1, \ldots\}$. Each time slot has a length of ζ, which is chosen to be sufficiently small to assume that the relative position between the satellite and the GW remains relatively stable during ζ. This allows us to analyze the system with a fixed communication channel. Consequently, UEs offload their workloads in two sequential stages. First, they transmit workloads to the GW via wireless links operating on the C-band above 4 GHz. Subsequently, the GW forwards them to the LEO satellite over the Ka-band above 20 GHz. After the computation is completed at the LEO satellite, the computing results are returned to the UEs in the reverse order. We assume that the computing results are so small that they can be considered negligible. We denote $\theta_i(t)$ as an offloading parameter of UE i at time slot t, represented by

$$\theta_i(t) = \begin{cases} 1, \text{ UE } i \text{ offloads its workload,} \\ 0, \text{ UE } i \text{ processes its workload locally, } \forall i \in \mathcal{I}. \end{cases} \quad (1)$$

2.2. Communication model

Each UE, each LEO, and GW are equipped with a single antenna,[1] and uses the orthogonal frequency-division multiple access (OFDMA) scheme [12]. The GW utilizes the C-band spectrum for frequency resources, which are evenly split and allocated to the UEs. Our system comprises two distinct communication processes: UEs-GW and GW-LEO satellite. The distance between UEs and the GW is much shorter than the GW-LEO satellite, resulting in a significantly better channel state. Consequently, the time required to exchange workloads is much less than that of the GW-LEO satellite link. Therefore, we focus on the GW-LEO satellite communication process for consistent analysis and assume that UEs-GW communication links are ideal with negligible latency, nearly zero compared to the GW-LEO satellite communication link.[2]

Using the Shannon capacity formula in wireless communication, we denote the achievable capacity of the GW-LEO link

[1] These configurations can be easily extended to multiple antennas by modifying the channel models and Eq. (2).

[2] It means that the GW-LEO link is always the bottleneck.

in the Ka-band during time slot t as follows.

$$r(t) = B_{Ka} \log_2 \left(1 + \frac{P_{gw}^N |h(t)|^2}{\hat{\sigma}^2}\right), \tag{2}$$

$$h(t) = v(t)\, A(t)\, \hat{d}(t)^{-\delta},\ \forall t \in \mathcal{T}, \tag{3}$$

where B_{Ka} represents the total bandwidth in the Ka-band, P_{gw}^N represents the transmit power of GW, $h(t)$ indicates the channel gain between GW and the LEO satellite at time slot t, $v(t) \sim Nakagami\,(m)$ signifies a Nakagami fading variable with parameter m at time slot t, $A(t)$ represents the rain attenuation variable in wireless communication systems at time slot t, $\hat{d}(t)$ indicates the distance between GW and the connected LEO satellite at time slot t, δ denotes the path loss exponent, and $\hat{\sigma}^2$ is the noise variance. We assume that the capacity of the link in each time slot is bounded as follows: $r(t) \le r_{\max}$.

2.3. Power dissipation model of computing and networking

Modern processors equipped with Dynamic Voltage and Frequency Scaling (DVFS) can dynamically adjust their CPU clock frequency, with processing speed denoted by $c(t) \in c_1(t), c_2(t), \ldots, c_{\max}(t)$ (in cycles/Δt) during time slot t. Each application has different computational requirements per bit. For example, chess games require significantly more computational resources per bit than image retrieval. We define processing density γ as the required CPU cycles to process one bit (in cycles/bit). The typical CPU power consumption model is described as follows.

$$P^p(c(t)) = \alpha_{u,1} c(t)^3 + \alpha_{u,2} c(t)^2 + \alpha_{u,3} c(t) + \alpha_{u,4},\ \forall t \in \mathcal{T}, \tag{4}$$

where $\boldsymbol{\alpha} = \{\alpha_{u,1}, \alpha_{u,2}, \alpha_{u,3}, \alpha_{u,4}\}$ is a set of parameters determined by the CPU model [15].

- **Mobile Device Power.** Let the power consumption of UE CPU processing at time slot t be $P_i^P(c_{u,i}(t))$ according to Eq. (4), where $c_{u,i}(t)$ represents the CPU clock frequency of UE i at time slot t. The total power consumption of a single UE includes both networking power for offloading and computational power for local processing. Thus, the overall power dissipation of UE i at time slot t is given as follows.

$$P_{u,i}(t) = \begin{cases} P_u^N, & \theta_i(t) = 1, \\ P_i^P(c_{u,i}(t)), & \theta_i(t) = 0,\ \forall t \in \mathcal{T}, \end{cases} \tag{5}$$

where P_u^N represents the transmit power of UE. Thus, the cumulative power dissipation of all UEs at time slot t is represented as follows.

$$P_{total,u}(t) = \sum_{i=1}^{I} \theta_i(t) P_u^N + \sum_{i=1}^{I} (1 - \theta_i(t)) P_i^P(c_{u,i}(t)). \tag{6}$$

- **LEO Satellite Power.** Unlike UEs, LEO satellite consumes only processing power in our system model. Therefore, the processing power consumption at time slot t in the LEO satellite is as follows [16].

$$P_s^P(c_s(t)) = \alpha_{s,1} c_s(t)^3 + \alpha_{s,2} c_s(t)^2 + \alpha_{s,3} c_s(t) + \alpha_{s,4},$$
$$= P_{total,s}(t),\ \forall t \in \mathcal{T}, \tag{7}$$

where $c_s(t)$ is the CPU clock frequency of the LEO satellite at time slot t. Despite the performance differences, it is observed that the DVFS model is applied similarly to both mobile device power and LEO satellite power.

- **Total Power.** As a result, the total power consumption at time slot t in the proposed system can be expressed as follows.

$$P_{sys}(t) = P_{total,u}(t) + P_{total,s}(t). \tag{8}$$

2.4. Workload queue model

On the ground layer, the ith UE generates $a_i(t)$ bits of workloads at the beginning of time slot t. During each time slot, a total of $\boldsymbol{a}(t) = (a_1(t), a_2(t), \ldots, a_I(t))$ workloads, measured in bits, arrive at each queue of each UE. It is important to note that $\boldsymbol{a}(t)$ is independent and identically distributed in every time slot, and its expected value is denoted as $\mathbb{E}[a_i(t)] = \lambda_u$. We assume that all arrivals in each time slot are bounded by a specified maximum value: $a_i(t) \le a_{\max}$.[3]

Unprocessed or unoffloaded workloads are stored in the each UE queue, provided there is sufficient storage capacity. As shown in Fig. 1, there are two types of workload queues (in bits) that evolve as follows.

$$Q_{u,i}(t+1) = \left[Q_{u,i}(t) - \frac{(1-\theta_i(t)) c_{u,i}(t)}{\gamma} \theta_i(t) r'(t) + a_i(t)\right]^+, \tag{9}$$

$$Q_s(t+1) = \left[Q_s(t) - N_c \frac{c_s(t)}{\gamma} + \sum_{i=1}^{I} \theta_i(t) r'(t)\right]^+, \tag{10}$$

where $Q_{u,i}(t)$ and $Q_s(t)$ represent the queue backlogs of the ith UE and the LEO satellite at time slot t, respectively, N_c is the number of CPU cores of LEO satellite, $r'(t)$ represents the maximum amount of workloads that a single UE can offload at a single time slot, and $[x]^+ = \max(x, 0)$.

3. Dynamic computation offloading and resource allocation algorithm

3.1. Problem formulation

In this section, we formulate an optimization problem to minimize power consumption and propagation delay constrained by queue stability. Due to the long distance between the LEO satellite and GW, we include propagation delay in the objective function as follows:

$$L_{sys}(t) = 2 \sum_{i=1}^{I} \theta_i(t) \frac{\hat{d(t)}}{l}, \tag{11}$$

where l denotes the speed of light, and $\hat{d(t)}$ indicates the distance between the LEO satellite and the GW at time slot

[3] We assume that the set of average arrived workloads is within the capacity region of the considered system, i.e., the system can stabilize the queues for any arrived workloads within the capacity region by an optimal service policy.

t. Thus, we present this long-term optimization problem as follows.

$$(\mathbf{P}): \min_{(\boldsymbol{\theta},\boldsymbol{c_u},\boldsymbol{c_s})} \left(\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\{P_{sys}(t) + wL_{sys}(t)\} \right), \tag{12}$$

$$s.t. \limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\left\{ \sum_{i=1}^{I} Q_{u,i}(\tau) + Q_s(\tau) \right\} < \infty, \tag{13}$$

where the control variables $(\boldsymbol{\theta}, \boldsymbol{c_u}, \boldsymbol{c_s}) \triangleq (\boldsymbol{\theta(t)}, \boldsymbol{c_{u(t)}}, c_s(t))_{t=0}^{\infty}$. Adjusting the weight w allows us to balance the tradeoff between system power consumption and propagation delay. Constraint (13) indicates that the average queue backlogs for each UE and the LEO satellite must be maintained within finite limits,[4] meaning that all arrived workloads should be served within a finite time [17].

3.2. Algorithm design

We derive the *DCOOL* algorithm using the Lyapunov drift-plus-penalty technique [17]. This method does not require knowledge of workload arrival distributions or future network states, transforming the stochastic optimization problem into an instantaneous optimization problem at each time slot.

- **Making slot-by-slot objective.** We first introduce the Lyapunov and Lyapunov drift functions as follows.

$$L(t) \triangleq \frac{1}{2} \left\{ \sum_{i=1}^{I} Q_{u,i}(t)^2 + Q_s(t)^2 \right\}, \tag{14}$$

$$\Delta(L(t)) \triangleq \mathbb{E}\{L(t+1) - L(t) \mid \mathbf{Q}(t)\}, \tag{15}$$

where $\mathbf{Q}(t) = \{Q_{u,1}(t), Q_{u,2}(t), \ldots, Q_{u,I}(t), Q_s(t)\}$. Lyapunov function (14) ensures the fair stability of all queues in UEs and LEO, maintaining similar queue backlogs. Minimizing the Lyapunov drift function (15) fairly minimizes the increment of each queue. We then introduce a Lyapunov drift-plus-penalty function, where the penalty function includes the expected system power consumption and propagation delay at time slot t, as detailed below.

$$\Delta(L(t)) + V\mathbb{E}\{P_{sys}(t) + wL_{sys}(t) \mid \mathbf{Q}(t)\}, \tag{16}$$

where V represents an objective-delay tradeoff parameter. In this context, "delay" refers to a processing delay related to the queue backlog, distinct from the propagation delay in the objective function. Therefore, the transformed single objective is to minimize the Lyapunov drift-plus-penalty function (16) at each time slot t.

- **Deriving an upper bound.** We establish an upper bound of Eq. (16) using queue evolution (9)–(10), workload arrivals and the objective function defined earlier.

Lemma 1. *Under any conceivable control variables* $\theta_i(t) \in (0, 1)$, $c_{u,i}(t) \in \{c_{u,1}(t), c_{u,2}(t), \ldots, c_{u,max}(t)\}$ *and* $c_s(t) \in \{c_{s,1}(t), c_{s,2}(t), \ldots, c_{s,max}(t)\}$, *we obtain:*

$$\begin{aligned} &\Delta(L(t)) + V\mathbb{E}\{P_{sys}(t) + wL_{sys}(t) \mid \mathbf{Q}(t)\} \\ &\le J + V\mathbb{E}\{P_{sys}(t) + wL_{sys}(t) \mid \mathbf{Q}(t)\} \\ &- \sum_{i=1}^{I} \mathbb{E}\left\{ \left(\frac{(1-\theta_i(t))c_{u,i}(t)}{\gamma} + \theta_i(t) r'(t) \right) Q_{u,i}(t) \mid \mathbf{Q}(t) \right\} \\ &- \mathbb{E}\left\{ \frac{N_c c_s(t)}{\gamma} Q_s(t) \mid \mathbf{Q}(t) \right\}, \end{aligned} \tag{17}$$

where $J = \frac{1}{2}\left(\sum_{i=1}^{I} \left\{ \frac{{c_{u,max}}^2}{\gamma^2} + {a_{max}}^2 \right\} + \frac{{N_c}^2 {c_{s,max}}^2}{\gamma^2} + 2{r_{max}}^2 \right)$.

Proof. We can prove it by following similar procedures with [13], yet we omit it due to space limitation. ■

Rather than tackling the complicated stochastic optimization problem described in Eq. (12), we suggest adopting the upper bound of (17) as the new objective function for minimization through a Lyapunov optimization-based approach [13]. Hence, the per-slot optimization problem can be redefined as

$$\begin{aligned} \min_{\boldsymbol{\theta}(t),\boldsymbol{c_u}(t),c_s(t)} \; & V\{P_{sys}(t) + wL_{sys}(t)\} - \frac{N_c c_s(t)}{\gamma} Q_s(t) \\ & - \sum_{i=1}^{I} \left(\frac{(1-\theta(t))c_{u,i}(t)}{\gamma} + \theta_i(t) r'(t) \right) Q_i(t). \end{aligned} \tag{18}$$

3.3. Dynamic COmputation Offloading and resource allocation for LEO satellite-assisted edge computing (DCOOL)

The DCOOL algorithm manages the offloading decisions, the CPU clock frequencies of UEs, and the CPU clock frequency of the LEO satellite $(\boldsymbol{\theta}(t), \boldsymbol{c_u}(t), c_s(t))$ for every time slot t. The goal of this algorithm can be characterized as finding a combination of control variables that minimizes the reformulated optimization problem (18). Taking a full exhaustive search to address this problem is overly complex and inefficient due to coupled data rate issues among UEs.[5] Hence, we propose a complexity-reduced version of the solution as follows.

(STEP 1) Find always offloading or local processing UEs : The reformulated problem (18), denoted as $U(t)$, can be divided into two main parts. The first part represents UEs engaged in offloading, denoted as $U_{i,off}(t)$, and those involved in local processing, represented by $U_{i,noff}(t)$. The second part

[4] In this formulation, we assume that the delay tolerance for each workload is uniform for simplicity. It is worth noting that we can accommodate various levels of delay tolerance by adjusting this constraint accordingly.

[5] Here, since all UEs share a single link among GW-LEO, achievable data rates for each UE change depending on the offloading decisions of each UE.

Algorithm 1 DCOOL Algorithm

Initialization
Initial input parameters γ, I, V, w, l, P_u^N *and* P_{gw}^N.
for $t = 1 : T$ **do**
  Given input parameters
  STEP 1: Find always offloading or non-offloading UEs.
  $\boldsymbol{\Theta(t) = \theta(:, t)}$
  for $i = 1 : I$ **do**
    if $\max U_{i,off}(t) < \min U_{i,noff}(t)$ **then**
      $\theta(i, t) = 1$.
    else if $\min U_{i,off}(t) < \max U_{i,noff}(t)$ **then**
      $\theta(i, t) = 0$.
    end if
  end for
  $\boldsymbol{\Theta_{\text{fixed}}(t) = \Theta(t)}$
  STEP 2: Optimize for remaining UEs.
  $U_{opt,u}(t) = \infty$.
  for every possible $\boldsymbol{\Theta_{\text{fixed}}(t)}$ **do**
    $S = \sum_{i=1}^{I} \theta(i, t)$, $r^{'}(t) = r(t)/S$.
    if $r^{'}(t) > r(t)$ **then**
      $r^{'}(t) = 0$.
    end if
    for $j = 1 : I$ **do**
      if $\theta(j, t) = 0$ **then**
        $c_{j,u}(t) = \underset{c_{u,j}(t)}{\operatorname{argmin}} U_{j,noff}(t)$
      end if
    end for
    $U_u^{'}(t) = U_u(\boldsymbol{\Theta_{\text{fixed}}(t), c_u(:, t)})$
    if $U_u^{'}(t) < U_{opt,u}(t)$ **then**
      $U_{opt,u}(t) = U_u^{'}(t)$, $\boldsymbol{\Theta_{opt}(t) = \Theta_{\text{fixed}}(t)}$.
      $\boldsymbol{c_{opt,u}(:, t) = c_u(:, t)}$, $r_{opt}^{'}(t) = r^{'}(t)$.
    end if
  end for
  $c_{opt,LEO}(t) = \underset{c_s(t)}{\operatorname{argmin}} U_{LEO}(t)$.
  STEP3: Update queue backlogs, $\forall i \in \mathcal{I}, s$.
end for

is unrelated to $\theta_i(t)$ and corresponds to the LEO satellite, denoted as $U_{LEO}(t)$. They are as follows.

$$U_{i,off}(t) = \sum_{i=1}^{I} \theta_i(t) \left(V \left\{ 2w \frac{d(t)}{l} + P_u^N \right\} - Q_{u,i}(t) r'(t) \right), \quad (19)$$

$$U_{i,noff}(t) = \sum_{i=1}^{I} (1 - \theta_i(t)) \left(V \left\{ \alpha_{u,1} c_{u,i}(t)^3 + \alpha_{u,2} c_{u,i}(t)^2 + \alpha_{u,3} c_{u,i}(t) + \alpha_{u,4} \right\} - Q_{u,i}(t) \frac{c_{u,i}(t)}{\gamma} \right), \quad (20)$$

$$U_{LEO}(t) = V \left(\alpha_{s,1} c_s(t)^3 + \alpha_{s,2} c_s(t)^2 + \alpha_{s,3} c_s(t) + \alpha_{s,4} \right) - N_c Q_s(t) \frac{c_s(t)}{\gamma}, \quad (21)$$

$$U(t) = \left(U_{i,off}(t) + U_{i,noff}(t) \right) + U_{LEO}(t). \quad (22)$$

Finding optimal decisions of code offloading and CPU clock frequency to minimize $U(t)$ is challenging due to the intertwined relations among control variables in $U(t)$. Therefore, we first compare the maximum and minimum values of the UE terms ((19), (20)) based on offloading decisions, seeking cases with significant differences. This helps identify whether UEs should always offload or process locally. Here, the selected UE 'always' performs the identified action in the current time slot. For instance, if the maximum offloading term of a UE (19) is significantly lower than the minimum local processing term (20), indicating that the objective value when offloading is much lower than when not offloading, the UE will always offload its workload to the LEO satellite without further comparison.

(STEP 2) Optimize for remaining UEs: For the group of UEs that do not determine control variables yet according to (STEP 1), we aim to identify the combination of control variables ($\boldsymbol{\theta}(t)$, $\boldsymbol{c_u}(t)$) that yields the smallest value of ((19)+(20)) among the possible offloading operation decisions. Notably, the objective function of the LEO satellite (21) remains uncorrelated with that of the UEs after (STEP 1). Consequently, we can independently optimize it to determine the optimal satellite CPU clock frequency $c_s(t)$.

(STEP 3) Update queue backlogs: After determining the control variables in (STEP 1) and (STEP 2), we update the amount of queue backlog for the next time slot according to the determined control variables. Finally, we repeat this procedure every time slot until the end.

3.4. Theoretical analysis

The sum of the queue backlogs and the average objective functions for UEs and LEO satellite achieved by the *DCOOL* algorithm can be upper bounded by Theorem 1 as follows.

Theorem 1. *Let $t = \{0, 1, \ldots, T-1\}$ and assume that there is a positive value $\epsilon > 0$ such that $\lambda_u + \epsilon \in \Lambda$, where Λ represents the capacity region[6] for arrival rates of UEs. Then, under the DCOOL algorithm, we have:*

$$\limsup_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} \left\{ \sum_{i=1}^{I} Q_{u,i}(t) + Q_s(t) \right\} \leq \frac{J + V E^*(\epsilon)}{\epsilon}, \quad (23)$$

$$\limsup_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} \{ P_{sys,D}(t) + w L_{sys,D}(t) \} \leq E^*(\epsilon) + \frac{J}{V}, \quad (24)$$

where sum of $P_{sys,D}(t)$ and $wL_{sys,D}(t)$ represents the objective function during time slot t when employing the proposed DCOOL algorithm. Meanwhile, $E^*(\epsilon)$ is the optimal lower bound of the objective function.

Proof. We can prove it by following similar procedures with [13], yet we omit it due to space limitation. ■

[6] This refers to the set of all arrival rates that can be transmitted or processed by the UE equipment within a finite time.

Table 1
Simulation parameters.

Parameter	Value	Parameter	Value
γ	1000 cycles/bit	I	10
w	150	B_{Ka}	100 MHz
N_c	8	$c_{u,max}$	3.4 GHz
P_u^N	2 W	$c_{s,max}$	4.5 GHz
P_{gw}^N	20 W	λ_u	2.5 Mbits

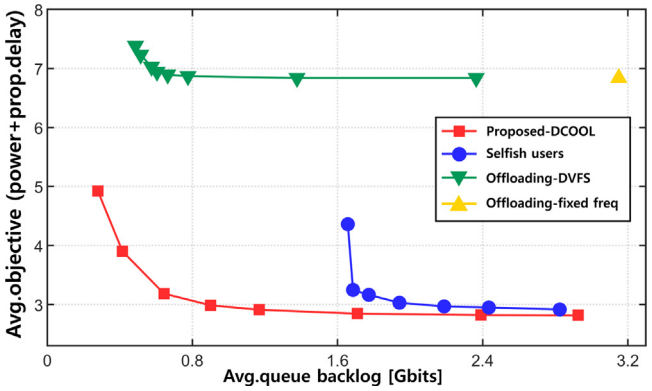


Fig. 2. Objective-queue backlog tradeoff.

Eqs. (23) and (24) illustrate the concept of an objective value-queue backlog tradeoff. When the parameter V decreases, the total average queue backlogs decrease while the average objective value increases, and vice versa. Additionally, these performance bounds can become tighter or looser depending on the value of J, which is influenced by the system environment.

4. Simulation results

4.1. Simulation settings

In this section, we present numerical results to assess the performance of the proposed *DCOOL* algorithm. We deploy a single satellite orbiting above the designated target area, with 550 km altitude and an orbital velocity of 7.66 km/s. The background noise density is set to be -174 dBm/Hz. For a comprehensive overview of the other critical parameters, please refer to Table 1 [18].

For comparison purposes, we evaluate several offloading and computation algorithms, including (1) the proposed *DCOOL* algorithm (*Proposed-DCOOL*), (2) an algorithm in which each UE optimizes their performance using *DCOOL* algorithm without consideration of the situation of other UEs (*Selfish users*), (3) an algorithm in which UEs always perform offloading and the satellite uses dynamic CPU DVFS algorithm (*Offloading-DVFS*), (4) an algorithm in which UEs always perform offloading but the satellite uses a fixed CPU clock frequency for processing (*Offloading-fixed freq*).

4.2. Comparison with existing algorithms

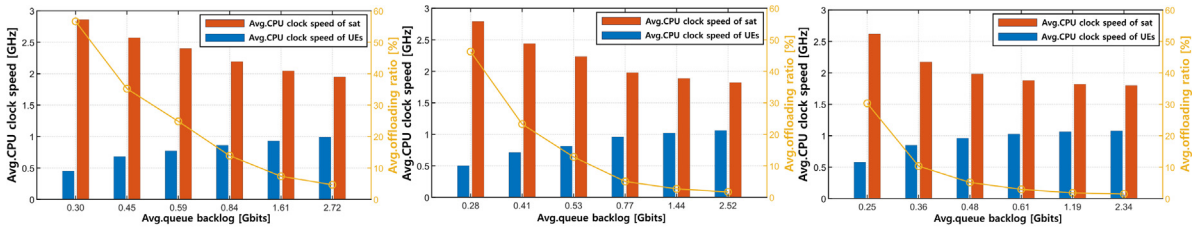
Fig. 2 illustrates the tradeoff between the objective value and queue backlog for *Proposed-DCOOL* and existing algorithms. The tradeoff, shown as a function of the Lyapunov tradeoff parameter V, indicates that increasing V decreases the average objective value but increases the average queue backlog. This variation is noticeable when V is not large enough (e.g., $V \leq 8 \cdot 10^{15}$), especially when the average queue backlog is below 0.8 Gbits. At higher V values, the tradeoff pattern is more gradual. The *Proposed-DCOOL* performs best when the average queue backlog is around 0.8 Gbits (i.e., $V = 8 \cdot 10^{15}$).

The *Proposed-DCOOL* algorithm shows remarkable performance compared to existing algorithms. It reduces the average objective value by 62% and 64% compared to the *Offloading-DVFS* and *Offloading-fixed freq* algorithms, respectively, for the same average queue backlog. The *Selfish users* algorithm performs slightly worse than *Proposed-DCOOL* within an average queue backlog range of 1.6 to 3 Gbits, but the performance gap increases significantly below this range. Thus, the *Proposed-DCOOL* algorithm operates more effectively in the small delay regime.

Compared to existing algorithms, the *Proposed-DCOOL* algorithm reduces objective values, particularly in system power consumption, due to several factors. These include: (1) The processing power consumption model changes rapidly with CPU clock frequency. Therefore, power consumption can be reduced by dynamically adjusting CPU clock frequency to maintain an appropriate queue backlog. (2) Wireless communication channel conditions affect the workloads sent in a single time slot, even with the same networking power. The rapid distance change between the LEO satellite and GW causes quick changes in wireless channel conditions. Waiting for better channel conditions to offload the same amount of workloads can lead to power-efficient transmissions. The *Proposed-DCOOL* algorithm considers the wireless communication channel conditions between the LEO satellite and GW at each time slot, enabling power-saving offloading decisions for each UE.

4.3. Impact of minimum altitude of LEO satellite

Fig. 3 illustrates the average CPU clock frequency of the LEO satellite and UEs, as well as the offloading ratio of UEs, for different minimum altitudes of the LEO satellite. As the minimum altitude increases, the quality of wireless channels between the LEO satellite and GW decreases, reducing offloading efficiency and the offloading ratio. Consequently, the workloads processed by the LEO satellite decrease, leading to a lower CPU clock frequency for the satellite. Conversely, workloads in UE queues increase, raising the CPU clock frequencies of the UEs. A common observation across different altitudes is that a similar ratio of control variables is selected when the average queue backlog exceeds a certain threshold (1.2 Gbits). This implies that the influence of the tradeoff parameter V becomes more significant than the wireless channel conditions at higher V values ($\geq 8 \cdot 10^{15}$).



(a) Minimum altitude: 150km (b) Minimum altitude: 550km (c) Minimum altitude: 950km

Fig. 3. Impact of the LEO satellite minimum altitude.

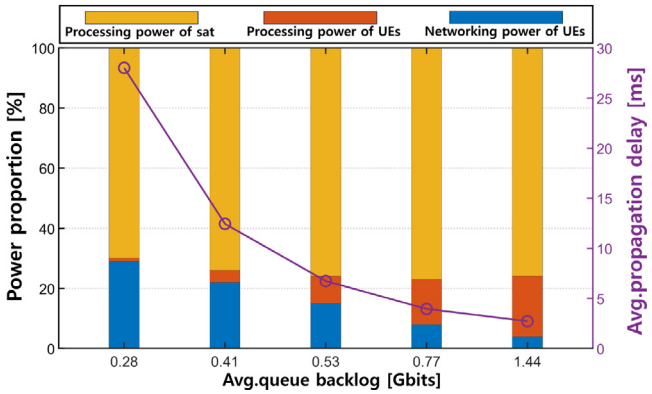


Fig. 4. Power proportion and propagation delay.

4.4. Impact of power proportion and propagation delay

Fig. 4 depicts the power consumption ratio and average propagation delay as a function of the average queue backlog. As the average queue backlog increases, the processing power consumption of the LEO satellite remains roughly constant, while that of the UEs increases. This observation can be related to Fig. 3 where the offloading ratio decreases as the average queue backlog increases, reducing the networking power consumption of UEs and the propagation delay. This demonstrates that the proposed algorithm adaptively controls CPU clock frequency and offloading policy in dynamic SMEC environments.

5. Conclusion

In this paper, we introduced a practical SMEC framework and proposed the *DCOOL* algorithm, which dynamically determines code offloading and CPU clock frequency of UEs and LEO satellite. Simulation results demonstrated that the proposed *DCOOL* algorithm outperforms the other existing algorithms in perspectives of average objective value and average queue backlog. It is worth mentioning that the system model used in this paper can be expanded to capture multiple LEOs and multi-terrestrial cells with a consideration of the association issue between UEs and LEO satellites. This can enhance the practicality of the research and is being considered as part of our future work.

CRediT authorship contribution statement

Jeonghwan Kim: Software, Validation, Writing – original draft, Writing – review & editing. **Jeongho Kwak:** Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2022-0-00704).

This work was supported by Korea Research Institute for defense Technology planning and advancement (KRIT) grant funded by the Korea government (DAPA (Defense Acquisition Program Administration)) (KRIT-CT-22-040, Heterogeneous Satellite constellation based ISR Research Center, 2022).

References

[1] J.P. Choi, C. Joo, Challenges for efficient and seamless space-terrestrial heterogeneous networks, IEEE Commun. Mag. 53 (5) (2015) 156–162.

[2] H.-C. Chao, D.E. Comer, O. Kao, Space and terrestrial integrated networks: Emerging research advances, prospects, and challenges, IEEE Netw. 33 (1) (2019) 6–7.

[3] T. Darwish, G.K. Kurt, H. Yanikomeroglu, M. Bellemare, G. Lamontagne, LEO satellites in 5G and beyond networks: A review from a standardization perspective, IEEE Access 10 (2022) 35040–35060.

[4] L. Boero, R. Bruschi, F. Davoli, M. Marchese, F. Patrone, Satellite networking integration in the 5G ecosystem: Research trends and open challenges, IEEE Netw. 32 (5) (2018) 9–15.

[5] I. Maity, G. Giambene, T.X. Vu, C. Kesha, S. Chatzinotas, Traffic-aware resource management in SDN/NFV-Based satellite networks for remote and urban areas, IEEE Trans. Veh. Technol. (2024).

[6] L. Zhao, C. Wang, K. Zhao, D. Tarchi, S. Wan, N. Kumar, INTERLINK: A digital twin-assisted storage strategy for satellite-terrestrial networks, IEEE Trans. Aerosp. Electron. Syst. 58 (5) (2022) 3746–3759.

[7] C. Wang, L. Zhao, C. Fan, K. Zhao, N. Kumar, J. Li, S. Wan, Metaverse-inspired cybertwin-based space-air-ground integrated networks, IEEE Netw. 37 (2) (2023) 294–300.

[8] N. Cheng, F. Lyu, W. Quan, C. Zhou, H. He, W. Shi, X. Shen, Space/aerial-assisted computing offloading for IoT applications: A learning-based approach, IEEE J. Sel. Areas Commun. 37 (5) (2019) 1117–1129.

[9] Z. Zhang, W. Zhang, F.-H. Tseng, Satellite mobile edge computing: Improving QoS of high-speed satellite-terrestrial networks using edge computing techniques, IEEE Netw. 33 (1) (2019) 70–76.
[10] C. Qiu, H. Yao, F.R. Yu, F. Xu, C. Zhao, Deep Q-learning aided networking, caching, and computing resources allocation in software-defined satellite-terrestrial networks, IEEE Trans. Veh. Technol. 68 (6) (2019) 5871–5883.
[11] Q. Tang, Z. Fei, B. Li, Z. Han, Computation offloading in LEO satellite networks with hybrid cloud and edge computing, IEEE Internet Things J. 8 (11) (2021) 9164–9176.
[12] X. Cao, B. Yang, Y. Shen, C. Yuen, Y. Zhang, Z. Han, H.V. Poor, L. Hanzo, Edge-assisted multi-layer offloading optimization of LEO satellite-terrestrial integrated networks, IEEE J. Sel. Areas Commun. 41 (2) (2022) 381–398.
[13] M.J. Neely, Stochastic network optimization with application to communication and queueing systems, Synth. Lect. Commun. Netw. 3 (1) (2010) 1–211.
[14] Y. Shi, Y. Cao, J. Liu, N. Kato, A cross-domain SDN architecture for multi-layered space-terrestrial integrated networks, IEEE Netw. 33 (1) (2019) 29–35.
[15] K. Son, B. Krishnamachari, Speedbalance: Speed-scaling-aware optimal load balancing for green cellular networks, in: Proc. of the 2012 IEEE INFOCOM, IEEE, 2012, pp. 2816–2820.
[16] Y. Zhang, H. Zhang, K. Sun, J. Huo, N. Wang, V.C. Leung, Partial computation offloading in satellite based three-tier cloud-edge integration networks, IEEE Trans. Wireless Commun. (2023).
[17] M. Neely, Stochastic Network Optimization with Application to Communication and Queueing Systems, Springer Nature, 2022.
[18] Y. Zhang, Q. Wu, Z. Lai, H. Li, Enabling low-latency-capable satellite-ground topology for emerging LEO satellite networks, in: IEEE INFOCOM 2022-IEEE Conference on Computer Communications, IEEE, 2022, pp. 1329–1338.