Received 1 May 2025; revised 2 June 2025; accepted 21 June 2025. Date of publication 1 July 2025; date of current version 25 July 2025.

Digital Object Identifier 10.1109/OJCOMS.2025.3584707

Enhancing Orbital Edge Computing Through ISL-Aided Federation of Satellite Constellations

S. PIZZI [1,2] (Member, IEEE), A. TROPEANO [1,2], G. ARANITI [1,2] (Senior Member, IEEE), AND A. MOLINARO [1,2,3] (Senior Member, IEEE)

[1]DIIES Department, University Mediterranea of Reggio Calabria, 89122 Reggio Calabria, Italy
[2]CNIT, 43124 Parma, Italy
[3]CentraleSupélec, Université Paris-Saclay, 91190 Gif-sur-Yvette, France

CORRESPONDING AUTHOR: S. PIZZI (e-mail: sara.pizzi@unirc.it)

This work was supported by the PRIN 2022 PNRR Project "ECO-AID: An Effective Monitoring Framework for Biodiversity Conservation that Combines ECOacoustics with Artificial Intelligence and Satellite Data Transmission and Elaboration" under Grant CUP: C53D23007230001.

ABSTRACT Future sixth-generation (6G) systems will undoubtedly be based on a three-dimensional (3D) network architecture encompassing jointly optimized terrestrial and non-terrestrial components. Non-Terrestrial Networks (NTNs) traditionally represent a key communication technology for complementing or replacing the insufficient or unavailable terrestrial infrastructure coverage in remote or disaster-affected areas. In such mission-critical scenarios, where data provided by massive Internet of Things (IoT) deployments needs to be analyzed quickly to make intelligent and rapid disaster rescue decisions, providing ubiquitous sensing, computing, and communication capabilities is essential to remotely monitor the area, control critical conditions, and take appropriate actions. Edge computing from the sky has recently become possible, enabling satellites at Low Earth Orbits (LEO) to provide on-orbit computing and storage capabilities in addition to large-scale connectivity. In this paper, a cooperative strategy is proposed that allows federations of LEO satellites, which are reachable via Inter-Satellite Links (ISLs), to share their computing, sensing, and storage resources. Specifically, satellites from constellations owned by different tenants are grouped into federations, where CubeSats equipped with regenerative payloads establish ISLs with nearby satellites and collaboratively perform sensing, computing, and storage tasks directly in orbit. The simulation results demonstrate the effectiveness of the proposed ISL-aided federation strategy in terms of the percentage of completed tasks, mean federation size, and mean waiting time under various satellite constellation settings (e.g., orbital altitude and number of satellites).

INDEX TERMS 6G, non-terrestrial networks, LEO satellites, IoT, cooperation, edge computing.

I. INTRODUCTION

PROVIDING connectivity from the sky has seen a resurgence of interest in recent years due to the effectiveness of satellite mega-constellations in achieving global and seamless network coverage [1]. Several commercial broadband satellite connectivity solutions are already available on the scene (e.g., Starlink, OneWeb, and Kuiper). Meanwhile, an increasing number of companies are pushing toward the satellite Internet of Things (IoT) to integrate satellite networks and IoT devices to enable seamless connectivity and data exchange. Thus, future sixth-generation (6G) networks will be based on a three-dimensional (3D) architecture encompassing jointly optimized terrestrial and non-terrestrial components [2].

While Terrestrial Networks (TNs) may offer limited and non-cost-effective coverage in specific scenarios, Non-Terrestrial Networks (NTNs) are key to realize truly global connectivity [3], providing network coverage in large domain areas and complementing terrestrial systems in un/underserved regions, thanks to their inherently large footprints.

A wide range of emerging services, such as high-definition video streaming, extended reality (XR), and holographic communication, which can be useful, for example, in telemedicine, remote diagnosis, patient monitoring and care, require the execution of computation-intensive and time-sensitive tasks, which IoT devices cannot accomplish due to limited computing and energy resources. A broadly adopted solution is to rely on ground-based data centers, taking advantage of the edge computing paradigm. However, in the case of insufficient or unavailable terrestrial infrastructure, especially in disaster scenarios, where collected data needs to be analyzed quickly to make rapid and intelligent rescue decisions, providing sensing, communication, and computing resources is essential to remotely monitor the area, control critical conditions, and take appropriate actions.

Recent innovations in device manufacturing have enabled Low Earth Orbiting (LEO) satellites to host on-orbit computing and storage capabilities. This realizes edge computing from the sky [4], which brings the significant benefits of reducing delays while being independent of the availability of terrestrial infrastructure. To further enhance satellite-edge computing capabilities, in our previous work [5], we proposed cooperation among LEO constellations of CubeSats to provide satellite-IoT services. Following this approach, satellites belonging to constellations owned by different tenants are organized in federations, wherein CubeSats' communication and computation resources are shared to enhance task execution and the overall system performance.

In this work, we make a step forward by exploiting another degree of freedom in space to further enhance the performance of IoT service provisioning. Indeed, LEO satellites embarking regenerative payloads can establish Inter-Satellite Links (ISLs) [6], which could be exploited to enlarge the availability of sensing, computing, and storage resources in the sky. The idea to propose an ISL-aided resource sharing strategy among multitenant LEO satellite constellations stems from the following main motivations:

- Miniaturized satellites, i.e., CubeSats, may be *constrained* in terms of computing/storage/sensing resources. Thus, a *cooperative approach* may enhance the overall task execution capability by offloading tasks in charge of one satellite constellation to other members of the federation, which may have unutilized resources or resource types that the initially assigned satellite constellation does not own.
- The *space-to-ground latency* can be critical for latency-sensitive applications (e.g., anomaly detection or emergency scenarios). For this reason, limiting space-to-ground communications by relying on an *in-space computation/storage/sensing system* may guarantee a timely service delivery.
- The resource availability may be further enhanced by exploiting not only the resources of the CubeSats in view with the edge node (i.e., the NTN Gateway, referred to as NTN-GW) but also those of their respective neighboring satellites by *exploiting ISLs*, thus fostering cooperation among satellite constellations and increasing the number of executed tasks.

These motivations lay the foundation of this paper, wherein we leverage the joint benefits of implementing cooperative approaches in a satellite-edge computing multi-tenant scenario and the possibility of utilizing the resources of neighboring satellites reachable via ISLs. As a result, we propose an ISL-aided federation strategy among satellite constellations aimed at realizing an effective computing center in the sky, thus improving task execution while reducing service delay.

The main contribution of this work can be summarized as follows:

- The design of an *ISL-aided federation strategy* for task offloading in a multi-tenant IoT scenario composed of LEO-based NTNs is presented by describing its main building blocks and their interactions. The design directly addresses the highlighted core motivations of our study.
- The model for *optimal CubeSat constellation cooperation*, building on the designed strategy, is provided to create federations of satellites wherein resources are shared among the members of the coalitions to enhance task execution capability of the system.
- A *heuristic solution* of the optimal strategy is introduced to enable practical implementation. It is based on coalition formation game for cooperation among CubeSat constellations, considering the personal preferences of the satellite tenants over the utilities they reach.
- A *comprehensive simulative campaign* is carried out to evaluate the benefits of the proposed ISL-aided federation strategy in terms of task execution capability and its flexibility to adapt to the needs of the satellite tenants.

This paper is organized as follows. In Section II, we set the background and the motivation behind this work. In Section III, we outline the scenario under investigation. The proposed ISL-aided federation strategy is described in Section IV. Performance results are analyzed in Section V, and conclusions are drawn in Section VII.

II. BACKGROUND AND RELATED WORK

Satellite networks have become affordable thanks to the development of commercial off-the-shelf (COTS) hardware and, consequently, to the relatively low costs and time from the development to the launch of CubeSats, particularly attractive to realize a global satellite network [7]. CubeSats are miniaturized satellites with a basic dimension of 10x10x10 cm^3 (1U) that promise global connectivity by realizing constellations at lower costs than traditional satellites. Transmissions from CubeSats to the NTN terminals exploit the S-band (2-4 GHz), C-band (4-8 GHz), X-band (8-12 GHz), Ku-band (12-18 GHz), or Ka-band (26.5-40 GHz). Furthermore, CubeSats are fed mainly by the sun as they are equipped with solar panels and move around the Earth's orbit at the altitude of LEO satellites, from 500 km onwards.

In the last years, CubeSats have been involved in new architectures based on emerging networking paradigms, such as the Internet of Space Things (IoST) [8] and the SoftSpace architecture with four segments (i.e., user, control-and-management, ground, and space) [9]. Applications, such as enhanced Mobile Broadband and Narrowband-Internet of Things (NB-IoT), have been considered in [10], where the impact of the satellite channel characteristics has been assessed on the physical and Medium Access Control layers. LEO satellites have a key role in backhauling in 5G networks [11] and are also part of the realization of Cloud-RAN for reliable wireless networks in complex areas in [12]. Several studies have focused on extending healthcare access, in particular telemedicine, to remote communities via satellite to improve accessibility in geographically isolated regions, from rural communities in Africa [13], [14], [15] and Canada [16].

Although LEO satellite constellations are currently used mainly to provide worldwide Internet access, the possibilities offered by the enhanced processing ability of LEO satellites, which can provide thousands Giga floating-point operations per second (GFLOPS) of computing capability, have pushed towards the investigation of LEO satellite edge computing [17]. Leveraging the computing capabilities of LEO satellites to implement an orbital computing continuum for equal access to computing is envisioned via simple scheduling techniques in [4]. Integrating edge computing into LEO networks is also analyzed in [18], where the functionalities that may be achieved in a LEO network's edge computing prototype system w.r.t. cloud computing are evaluated. Security issues in computation offloading to satellites are discussed in [19], where a security-aware offloading algorithm is designed by utilizing reinforcement learning. A satellite mobile edge computing (SMEC) framework for real-time and very-high resolution Earth observation is formulated in [20] to optimize the image distribution and compression parameters to minimize energy consumption. In [21], a distributed service deployment framework in a satellite edge computing network is proposed by jointly optimizing the allocation of computational, communication, and storage resources while minimizing the total service processing delay of time-sensitive applications.

Multi-tier edge computing architectures have been investigated in several research works [22], [23], [24]. A three-tier computation architecture that includes ground users, LEO, and cloud servers is proposed in [22]. Specifically, computation tasks generated by ground users can be computed locally, at LEO equipped with MEC, or on cloud servers via LEO offloading. The decision on where to offload the tasks is dictated by the strategy that minimizes the total energy consumption of ground users. A satellite-terrestrial edge computing architecture is proposed in [23], where principles and functionalities for deploying MEC in satellite-terrestrial networks are investigated. In [24], a joint communication and computation resource allocation problem is formulated in a LEO satellite edge-assisted system, aiming at minimizing the overall energy dissipation. The possibility of exploiting multi-layer satellite-based networks to support intelligent transportation systems nodes, such as airplanes and ships, is envisioned in [25], wherein the authors focus on optimizing the use of communication bandwidth and computing resources of CubeSats, LEO satellite-based MEC servers, and GEO satellites that function as core network servers.

The exploitation of cooperative approaches in satellite systems can bring meaningful benefits in a wide range of application scenarios. Federated satellite systems have introduced benefits due to the possibility of sharing resources between satellites to maximize the system utility [26]. A 5G satellite-IoT platform empowered with virtualization and edge computing capabilities is designed in [5] where virtualized CubeSat constellations cooperate and federate under terrestrial edge nodes' control to provide IoT services. In [27], a cooperative multilayer edge caching solution is designed to reduce the communication delay in the integrated satellite-terrestrial network, in which the base station cache, the satellite cache, and the gateway cache cooperatively provide content service for ground users. In [28], a cooperative edge-cloud offloading problem is optimized to minimize the total delay for an integrated satellite-terrestrial network, and the optimal task splitting strategy is derived with a closed-form solution. Taking advantage of the digital twin (DT) technology, in [29], the authors propose a digital twin-empowered satellite-terrestrial cooperative computing framework composed of terrestrial users, base stations, LEO satellites, and a cloud center. The computation tasks from terrestrial users can be partially offloaded to the BS edge server, the associated LEO satellite edge server, and the adjacent LEO satellite edge server.

A. NOVELTY OF THE WORK

The above review highlights that there is an intensive research activity focused on deploying tasks to satellites for on-orbit processing. However, due to limitations in space computational power, communication capacity, and discontinuous ground-to-space visibility due to satellite movement, excessive overhead for data transmission and processing make task offloading challenging and particularly critical in the case of large-scale satellite networks and time-critical computation-intensive service delivery.

Moreover, we consider the multi-tenancy issue, recognizing that the satellite network in the sky can be composed of CubeSats equipped with heterogeneous payloads, and belonging to different constellations owned by different tenants. We focus on a multi-tenant IoT satellite scenario, composed of space nodes with on board computing/sensing/storage resources and, extending our prior work in [5], we design a federation strategy among satellite cloud providers that leverages the possibility of utilizing the resources of neighboring satellites reachable via ISLs to realize an effective computing center in the sky, thus improving task execution while reducing service delay.

III. SCENARIO DESCRIPTION

We consider the multi-tenant IoT satellite scenario shown in Fig. 1. Specifically, we define a *Satellite Cloud Provider (SCP)* as a CubeSat constellation that is a set of CubeSats owned by the same SCP. The scenario includes an *edge*

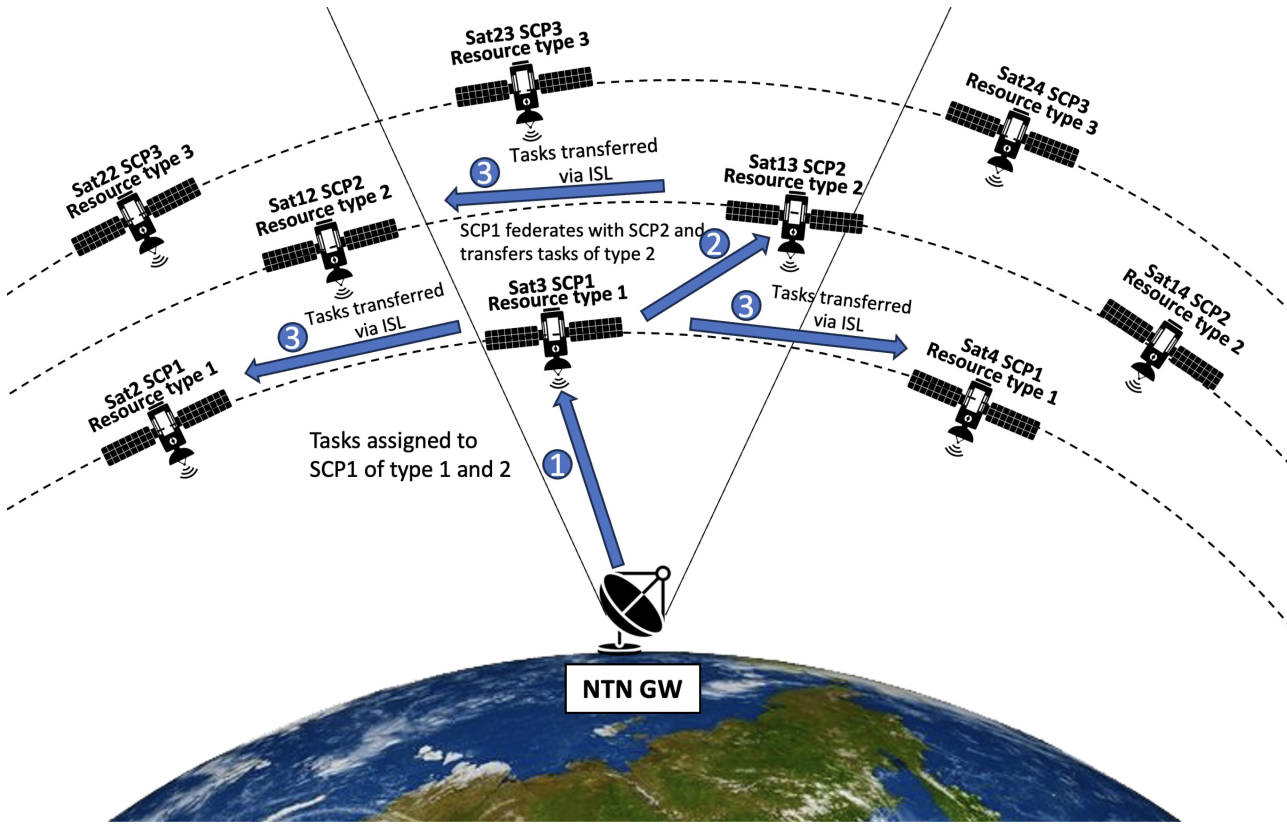


FIGURE 1. **Example multi-tenant IoT satellite scenario.**

node, that is the NTN-GW, which has the main objective of creating and orchestrating the SCPs so that the number of executed tasks is maximized. When the SCP registers the CubeSats at the edge node, an *SCP controller* responsible for managing available resources (e.g., storage, computation, and sensing) is created at the NTN-GW.

Since CubeSats follow a precise trajectory along their orbit according to mathematical laws, they can only connect to the NTN-GW in specific time intervals that can be predicted. We consider the *federation time* as the time during which a CubeSat is in Line-of-Sight (LoS) with the NTN-GW, which performs federation selection and, accordingly, task allocation considering, for each SCP, the resources owned by the CubeSats in view and those of nearly CubeSats reachable with h hops via ISLs.

An SCP is visible from an NTN-GW if one or more of its satellites are above the local horizon, ideally with an elevation angle higher than 10–15°, which is typically needed for a reliable radio link. Among the key factors that affect space-to-ground visibility are the satellite altitude, orbital distribution, NTN-GW location, and constellation density. Based on the actual coverage provided by companies of the LEO satellite market, we can estimate that, on average, 3 to 5 LEO constellations can be simultaneously visible from an NTN-GW at any given moment.

We assume that satellites belonging to an SCP are placed on a circular orbital plane, i.e., each plane $i \in \mathcal{I}, \mathcal{I}=\{1, 2, \ldots, I\}$ consists of S_i uniformly distributed satellites, $\mathcal{S}_i=\{s_1, s_2, \ldots, s_{S_i}\}$, characterized by an altitude H_i, inclination angle A_i, and orbital period T_i.

We assume that the NTN-GW can request the execution of n different types of tasks, $n \in \mathcal{N}, \mathcal{N}=\{1, 2, \ldots, N\}$, e.g., different kinds of sensing activities, storage, or computing. Tasks are randomly assigned to the SCPs in view. Each satellite owns r resources, $r \in \mathcal{R}, \mathcal{R}=\{1, 2, \ldots, R\}$. Each resource r is tailored to one of the types of tasks in $\mathcal{N}$.

This association can be introduced through the function $f : \mathcal{N} \rightarrow \mathcal{R}$. For example, one satellite has three types of resources: type r_1 for sensing, type r_2 for storage, and type r_3 for computing. In case tasks of all these three types are assigned to the satellite (task t_1 of type sensing, task t_2 of type storage, task t_3 of type computing), the satellite can execute all assigned tasks since $f(t_1)=r_1$, $f(t_2)=r_2$, and $f(t_3)=r_3$.

SCPs are assumed to be homogeneous (i.e., satellites belonging to the same SCP own the same resources). Consequently, satellites may not own the resources required to accomplish the requested tasks.

IV. PROPOSED ISL-AIDED FEDERATION STRATEGY FOR TASK OFFLOADING IN LEO-BASED NTNS

The working logic of the proposed solution is sketched in Fig. 2, where all the main building blocks and the interactions among them are depicted.

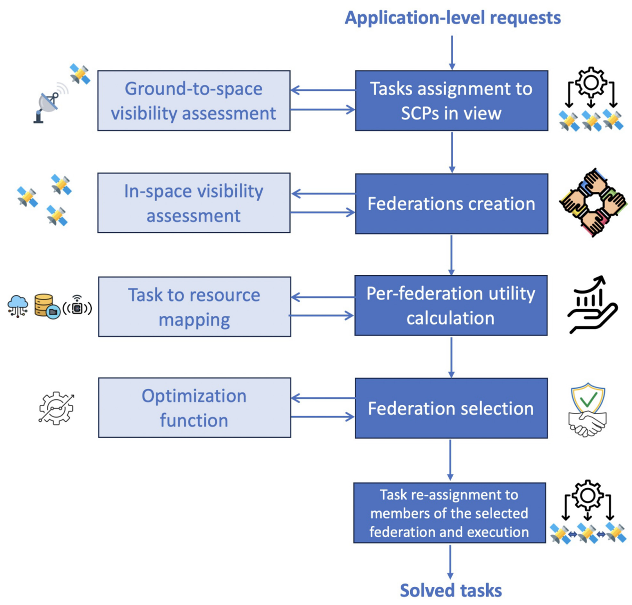


FIGURE 2. Block diagram illustrating the working logic of the proposed solution.

- *Tasks assignment to the SCPs in view:* Starting from application-level requests, the edge node (i.e., the NTN GW) assigns the tasks to the SCPs with at least one CubeSat in view identified by checking the ground-to-space visibility.
- *Federations creation:* All possible federations among the SCPs are computed after evaluating the in-space visibility to allow CubeSat-to-CubeSat communication between satellites in the federations for task offloading.
- *Per-federation utility calculation:* For each possible federation, the utility is calculated considering the total number of tasks assigned and its ability to execute those tasks based on their type and the availability of the resources required to execute the overall tasks assigned in the federation. Specifically, tasks are assigned to the SCP that owns the resources to solve them, and tasks of the same type are executed sequentially by exploiting a single resource of the same type as the task. We highlight that the task allocation algorithm considers resources owned by a federation to be also those of other CubeSats that are n hops away from the CubeSats in view of the edge node reachable via ISLs.
- *Federation selection:* Among all possible federations, the algorithm selects the one that allows the utility function to be maximized, which is detailed in the next subsection.
- *Task re-assignment:* Tasks are re-assigned to the members of the selected federation and executed by the CubeSats that own the resources required to solve them.

In Fig. 1, an example scenario is depicted that shows the interactions between NTN-GW and SCP, intra-SCP, and inter-SCPs. The NTN-GW is assumed to assign tasks of types 1 and 2 to SCP1. Thus, since SCP1 owns only resources of type 1, it federates with SCP2, which transfers tasks of type 2. Then, tasks are transferred by the CubeSats of both SCP1 and SCP2 to their neighboring CubeSats via ISLs to exploit the resources of nodes that are not in view, thereby improving service delivery.

The proposed model assumes that task exchange and cooperation occur within a limited neighborhood, typically within one or two hops. This design choice deliberately avoids the complexities of end-to-end routing across the broader LEO network, allowing us to focus on local cooperation mechanisms that are more practical and feasible in near-term deployments. Routing in large-scale LEO constellations remains an active area of research, particularly for dynamic and delay-sensitive applications. However, in our scenario, the localized nature of the federation and the proximity of collaborating satellites simplify the routing challenge, reducing it to short-range task forwarding, which can be managed with relatively straightforward coordination policies.

Another important aspect that our proposal takes into account is that spaceborne computational resources are limited. We explicitly address this constraint by enabling task sharing across multiple satellites. Indeed, rather than relying on the processing capabilities of a single node, tasks are distributed across neighboring satellites in the federation, thereby pooling their limited computing power to meet the application's requirements. This cooperative approach mitigates individual resource limitations and enhances the overall system's ability to execute more demanding tasks than would be possible on isolated nodes.

Moreover, our ISL-aided federation strategy is specifically designed to mitigate congestion by enabling distributed task execution across satellite nodes from different constellations. By balancing the load across multiple resources, we expect the federation approach to outperform non-federated systems even in congested conditions. In particular, the proposed solution relies on a set of inputs, including available satellite resources, application requirements, and satellite visibility. Most of these inputs are available or predictable at the edge node (i.e., the NTN-GW). Once a federation is formed, tasks are offloaded to its members via ISLs, introducing some communication overhead. This overhead is task-dependent: for instance, a sensing task may only require a simple request control message, while a computing task may involve transmitting larger data volumes. It is important to note that ISLs are increasingly capable, with technologies such as RF and laser-based links offering high data rates. These advancements suggest that the overhead introduced by our solution is well within the capabilities of current and upcoming satellite systems and does not represent a barrier to its practical implementation.

As a final remark, we underline that the proposed solution is still applicable even in the case an NTN-GW is not available (e.g., far in the sea or remote areas) since any CubeSat can play the role of edge node. However, due to the mobility of satellites, the edge node election is a dynamic procedure that requires proper design. In addition, we wish to highlight that in the case of applications such as anomaly detection, exploiting the proposed solution may be significantly beneficial since cooperation leads to meaningful advantages in task execution capability and response time, which is a critical parameter in such scenarios.

A. MODEL FOR OPTIMAL CUBESAT CONSTELLATIONS COOPERATION

The cooperation of CubeSat constellations is modeled by determining all possible coalitions $\mathcal{F}=\{\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_F\}$ among the I SCPs. The total number of such coalitions F is calculated as

$$F = \bigcup_{k=1}^{I} C(I,k) = \bigcup_{k=1}^{I} \frac{D(I,k)}{P(k)} = \\ = \bigcup_{k=1}^{I} \frac{I \cdot (I-1) \cdot (I-2) \cdot \ldots \cdot (I-k+1)}{k!}, \quad (1)$$

where $C(I, k)$ is the number of k-combinations without repetitions of the I SCPs, $D(I, k)$ is the number of k-permutations of the I SCPs, $k<=I$, and $P(k)$ is the number of permutations of the k SCPs. As an example, in the case of 4 SCPs (i.e., $I=4$), the total number of coalitions F is 15, which is given by the union of the unique combination of $k = 4$ elements (i.e., {1,2,3,4}), four combinations of $k = 3$ elements (i.e., {1,2,3}, {1,2,4}, {2,3,4}, {1,3,4}), six combinations of $k = 2$ elements (i.e., {1,2}, {1,3}, {1,4}, {2,3}, {2,4}, {3,4}), and four combinations of $k = 1$ elements (i.e., {1}, {2}, {3}, {4}).

For each possible coalition $\mathcal{F}_f \in \mathcal{F}$, we first determine the utility of each member i_i participating in coalition $\mathcal{F}_f$, referred to as *individual utility* $u_{i,\mathcal{F}_f}$. Then, we determine the *coalition utility* $u_{\mathcal{F}_f}$ of the coalition as an average value of the individual utility of all SCPs participating in the coalition, that is

$$u_{\mathcal{F}_f} = \frac{\sum_{i=1}^{|\mathcal{F}_f|} u_{i,\mathcal{F}_f}}{F}. \quad (2)$$

Among the coalitions in $\mathcal{F}$, the coalition that provides the highest utility, $\mathcal{F}_f^*$, is selected, that is

$$\mathcal{F}_f^* | u_{\mathcal{F}_f}^* = \max_{\mathcal{F}_f \in \mathcal{F}} u_{\mathcal{F}_f}. \quad (3)$$

Utility expresses the advantage that an SCP has in federating with other SCPs and is given by the difference between the *benefit* $b_{i,\mathcal{F}_f}$ obtained and the *cost* paid $c_{i,\mathcal{F}_f}$. Specifically, the utility $u_{i,\mathcal{F}_f}$ of member i participating in coalition $\mathcal{F}_f$ is calculated as

$$u_{i,\mathcal{F}_f} = b_{i,\mathcal{F}_f} - \gamma_i \cdot c_{i,\mathcal{F}_f}, \quad (4)$$

where γ_i is a parameter expressing the attitude of an SCP to participate in a coalition, referred to as *compromise factor*. It is set per SCP and takes values in the range [0, 1], where the value 0 indicates the willingness of an SCP to federate, while the value 1 suggests a non-cooperative behavior.

The *benefit* $b_{i,\mathcal{F}_f}$ of member i participating in coalition $\mathcal{F}_f$ is calculated as the number of successfully executed tasks of all type n (executed either by SCP i, $s_{i,n}^{ex}$, or by other SCPs j participating in the coalition $\mathcal{F}_f$, $s_{j,n}^{ex}$) over the number of tasks assigned to SCP i, $s_{i,n}^{ass}$, that is:

$$b_{i,\mathcal{F}_f} = \frac{\sum_{n=1}^{N} \left(s_{i,n}^{ex} + \sum_{j \in \mathcal{F}_f, j \neq i} s_{j,n}^{ex} \right)}{\sum_{n=1}^{N} s_{i,n}^{ass}}. \quad (5)$$

The *cost* $c_{i,\mathcal{F}_f}$ of member i participating in coalition $\mathcal{F}_f$ is calculated as the number of successfully executed tasks initially assigned to other SCPs participating in the coalition $\mathcal{F}_f$ over the total number of executed tasks, that is:

$$c_{i,\mathcal{F}_f} = \frac{\sum_{n=1}^{N} \sum_{j \in \mathcal{F}_f, j \neq i} s_{j,n}^{ex}}{\sum_{n=1}^{N} s_{i,n}^{ex}}. \quad (6)$$

The reasoning behind this setting for the utility function is that an SCP takes advantage when a high percentage of assigned tasks is executed by exploiting the resources of other members of the federation (i.e., the benefit) at the expense of utilizing its owned resources at the advantage of other SCPs in the federation (i.e., the cost).

B. A COALITION FORMATION GAME FOR SCPS COOPERATION

In this section, we model the cooperation among SCPs using coalitional game theory. We define a non-transferable utility coalition game characterized by the tuple $(N, X, V, (\succ_i)_{i \in N})$, where N is a set of players (i.e., the SCPs), X is a set of payoffs, V is a function that describes the outcomes $V(C) \subseteq X$ that can be achieved by coalition C, and $\succ_i$ is the preference relation of player i over the set of payoffs. The game is in the *characteristic form* because the utility that a coalition can achieve depends only on the players in that coalition.

In this coalition game, the objective of the players is to maximize their utility in the coalition they belong to. The utility is computed as in Eq. (4), where the parameter γ_i tracks the rationality of the players in the game and can be dynamically tuned according to the preference of the SCPs.

Since the presence of a cost term in the utility function in Eq. (4) may lead to a non-superadditive game with disjoint coalitions, to avoid such effect, SCPs may decide to join or leave a coalition on the basis of well-defined preferences. Specifically, we define a preference order (i.e., $\succ_i$) for any SCP $i \in N$ over the set of coalitions to which SCP i can belong. Specifically, we say that SCP i prefers coalition C_i to C_i' if $C_i \succ C_i'$. To compare two partitions, we define the preference order as follows:

$$C_i \succ C_i' \Leftrightarrow f(C_i) > f\left(C_i'\right) \quad (7)$$

where, for any SCP i and coalition C, the preference function $f(C_i)$ is defined as follows:

$$f(C_i) = \begin{cases} v_i(C_i), \; C_i \notin H(i) \vee \mid C_i \mid = 1; \\ -\infty, \quad otherwise. \end{cases} \quad (8)$$

Algorithm 1 Formation of SCPs Federations

Input: Set of SCPs N
1: $P^i = N = \{\{1\}, \{2\}, \dots, \{N\}\}$;
2: $H(i) = \emptyset\ \forall i \in N$;
3: $P^c = \{\{C_1\}, \{C_2\}, \dots, \{C_l\}\} = P^i$;
4: **repeat**
5: **for** $i \in N$ **do**
6:   **for** $C_k \in P^c$ **do**
7:     **if** $C_k \bigcup\{i\} \succ C_i$ **then**
8:       $P' = P^c \setminus \{C_i\} \bigcup \{C_i \setminus \{i\}, C_k \bigcup \{i\}\}$;
9:       $H(i) = H(i) \bigcup C_i$;
10:       $P^c = P'$;
11:     **end if**
12:   **end for**
13: **end for**
14: **until** $P^f = P^c$
15: **Output:** Final partition $\mathcal{P}^f$

In detail, $v_i(C_i)$, $v_i \in V$, is the payoff value for the i-th SCP in the coalition C_i, while $H(i)$ stores the coalitions that the algorithm has already considered to avoid the same coalition being examined twice [31].

Based on this preference order, the coalition formation employs a *switch operation*. In particular, given a partition $P = \{C_1, \dots, C_l\}$ on the set of players N and the preference relation in Eq. (8), an SCP i decides to switch coalition and leave the current coalition to join another coalition only if it prefers to be part of the new coalition according to the preference relation.

Algorithm 1 lists the pseudo-code for the coalition formation game for SCPs cooperation.

For the coalition creation, the algorithm receives the set N of SCPs as input, creates an initial partition P^i, where each SCP represents a distinct coalition, and initializes the history to an empty set as no coalition has been analyzed yet (lines 1-2). Then, the algorithm considers the preference order to determine the execution of the switch operation. Specifically, for each coalition C_k in the current partition $P^c = \{C_1, C_2, \dots, C_l\}$ on the set N of SCPs (i.e., players) and the preference relation defined in Eq. (8), an SCP $i \in N$ switches from coalition C_i to C_k only if $C_k \bigcup\{i\} \succ C_i$ (lines 3-8). Once an SCP i decides to be part of the new coalition C_k, the history $H(i)$ is updated (line 9). Similarly, the current partition P^c, which is initially formed by coalitions with single SCPs (i.e., the cardinality of P^i is equal to the number of SCPs), is revised (line 10) until it converges to the final partition P^f, thus interrupting the switch operation (lines 4-14). The algorithm provides the final partition of the SCPs federation (line 15).

C. COMPUTATIONAL COMPLEXITY

In our proposed approach, the federation formation process is performed by the edge node–specifically, the NTN-Gateway (NTN-GW)–which is not subject to the computational and energy constraints typical of satellite nodes. This centralized design serves two main purposes: (i) it minimizes the computational burden on the resource-constrained satellite nodes, and (ii) it leverages the global view of the network available at the edge node to improve decision-making.

The computational complexity of our proposed strategy, as described in Algorithm 1, can be summarized as follows:

- *Initialization phase (lines 1-3):* This step has a complexity of $O(N)$, where N is the number of SCPs.
- *Main loop (lines 4-13):* In the worst-case scenario, each SCP could switch coalitions at most once for each possible coalition it could join. Since there are at most $2^N - 1$ coalitions, the number of iterations is bounded by $O(N \cdot 2^N)$.
- *Per iteration operations:* For each SCP, we evaluate potential membership in each coalition of the current partition. Since the number of coalitions in any partition is at most N, this operation has a per-iteration complexity of $O(N^2)$.
- *Preference evaluation (line 7):* Computing preference involves calculating utilities as defined in Eq. (4), which scales with the number of task types T, yielding a complexity of $O(N \cdot T)$.

Combining these, the overall worst-case computational complexity of the algorithm is $O(N^4 \cdot 2^N \cdot T)$.

However, we emphasize that this represents a theoretical upper bound. In practical scenarios:

- The number of SCPs (N) is relatively small, as it corresponds to the number of satellite constellations visible from the edge node at any given time, making the exponential term manageable.
- Convergence is significantly faster in practice. Our simulation results show that the algorithm typically converges within a few iterations.
- The algorithm runs at the NTN-GW, which has sufficient processing resources, ensuring runtime is not a bottleneck.

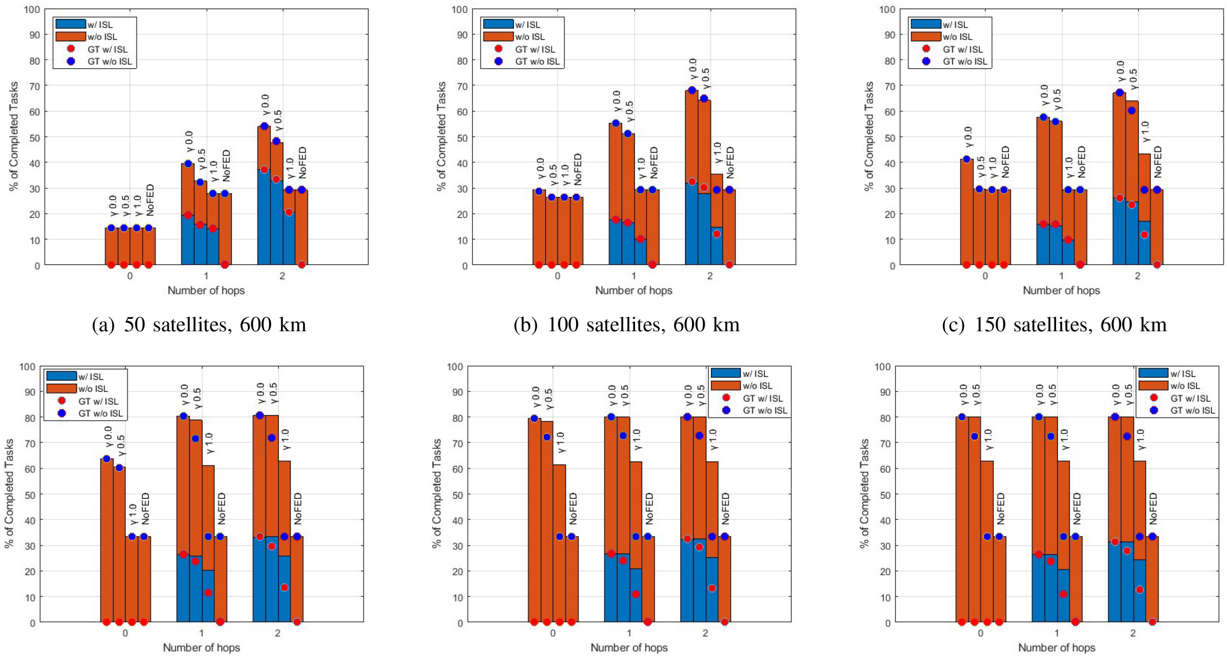
These factors ensure that the proposed strategy remains scalable and practical for real-world deployment scenarios.

V. SIMULATION RESULTS

Simulations have been carried out using an ad-hoc developed MATLAB simulator, which allows deploying satellite constellations given the number of satellites per constellation, the satellite orbital height, the inclination angle, and the coverage angle.

The scenario under investigation comprises four SCPs, each consisting of 50 to 150 satellites, and one NTN-GW located on the ground by indicating latitude and longitude coordinates. CubeSats carry regenerative payloads and orbit at 600 or 1200 km around Earth, thus communicating with the NTN-GW in a time equal to 12.89 and 20.89 ms, respectively [30]. For each task transferred over ISL, we accounted for propagation delays based on satellite altitude, ensuring a realistic evaluation of performance.

Tasks are randomly generated of three different types. Satellite constellations are homogeneous, i.e., all satellites



(a) 50 satellites, 600 km (b) 100 satellites, 600 km (c) 150 satellites, 600 km

(d) 50 satellites, 1200 km (e) 100 satellites, 1200 km (f) 150 satellites, 1200 km

FIGURE 3. **Percentage of completed tasks for an orbital altitude of 600 km ((a), (b), and (c)), and 1200 km ((d), (e), and (f)). Results are shown for a number of satellites equal to 50 ((a) and (d)), 100 ((b) and (e)), and 150 ((c) and (f)).**

belonging to the same constellation have resources of the same type, which is randomly generated of one out of the three different types required to solve the tasks. Each satellite has only one resource. This assumption on homogeneity among satellites within an SCP is based on the fact that satellites within a given constellation are often launched with a unified purpose (e.g., Earth observation, communication) and are typically designed to be identical in terms of hardware and capabilities. Therefore, while the assumption simplifies the model, we believe it reflects a common deployment practice in real-world satellite constellations. Differently, the resource types assigned to different SCPs are randomly generated from three distinct task types, ensuring that the system captures a realistic variation in capabilities across constellations. This heterogeneity is leveraged to enhance federation formation and improve task execution through resource complementarity. Furthermore, we note that while our current implementation assumes intra-constellation homogeneity, the proposed federation formation strategy is flexible and can be extended to accommodate intra-constellation heterogeneity as well.

Table 1 summarizes the main simulation parameters.

The analyzed metrics of interest are:

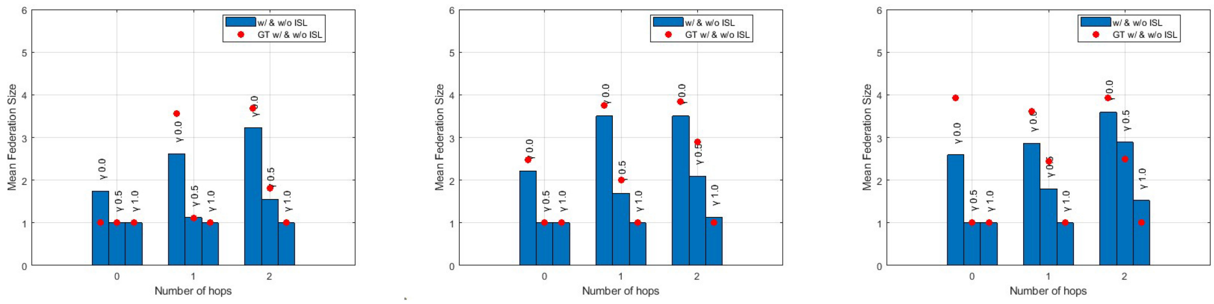
- *Percentage of completed tasks*, calculated as the ratio between the total number of tasks executed and the total number of tasks assigned by the edge node.
- *Mean federation size*, representing the average number of SCPs in the selected federations.
- *Mean waiting time*, accounting for the average time that elapses between the instant of time a task is assigned and the time it is executed (including propagation, queuing, and execution delays).

TABLE 1. **Main simulation parameters.**

Parameter	Value
Number of SCPs	4
Number of satellites per SCP	50, 100, 150
Orbital height [Km]	600, 1200
Number of resources per satellite	1
Number of resource types	3
Number of task requests	2000, 3000, 4000
Task execution time	1 s
Number of runs per simulation	50
Compromise factor	0, 0.5, 1

Results achieved by implementing the optimal federation strategy are shown as bars in Section IV-A, whereas we display as points the performance of the game theoretical (GT) solution presented in Section IV-B. The performance of the proposed solution is analyzed in comparison to the baseline approach that does not consider federation among SCPs, labeled as “No FED” in the presented plots.

Figure 3 analyses the percentage of tasks completed in different configurations of satellite constellations, specifically focusing on the impact of ISLs on task completion efficiency. For this metric, we distinguish between the percentage of tasks solved by the satellite in view with the NTN-GW (i.e., w/o ISL, red bars) and those executed by satellites reached via ISLs (i.e., w/ ISL, blue bars). From the plots, we can infer that the proposed federation policy allows for a remarkable improvement compared to no federation in case the SCPs are prone to share their resources (i.e.,



(a) 50 satellites, 600 km

(b) 100 satellites, 600 km

(c) 150 satellites, 600 km

FIGURE 4. **Mean federation size for an orbital altitude of 600 km for a number of satellites equal to 50 (a), 100 (b), and 150 (c).**

$\gamma = 0$), especially for higher orbital altitudes. In fact, for 150 satellites per constellation orbiting at 1200 km and considering the possibility of exploiting the resources of satellites two hops via ISLs far from the satellites in view (see Figure 3(f)), the percentage of completed tasks increases from about 30% to about 80%. By decreasing the compromise factor, we underline that the task completion capability drops and reaches values close to the no-federation policy under a non-cooperative attitude of the SCPs (i.e., $\gamma = 1$).

The reason behind the higher performance achieved under a highly cooperative attitude of the SCPs (i.e., $\gamma = 0$) is the creation of federations of SCPs of increased size. In fact, as can be inferred from Figure 4, lower values of γ lead to higher mean federation size values due to the lower weight of the cost term in the calculation of the per-SCP utility in Eq. (4). Additionally, we note that the mean federation size increases with the number of satellites per constellation, due to the higher availability of resources. Finally, we observe that, in most cases, GT achieves slightly higher values of the mean federation size. These results suggest that while the heuristic strategy tends to form larger federations, it may not consistently enhance task execution performance.

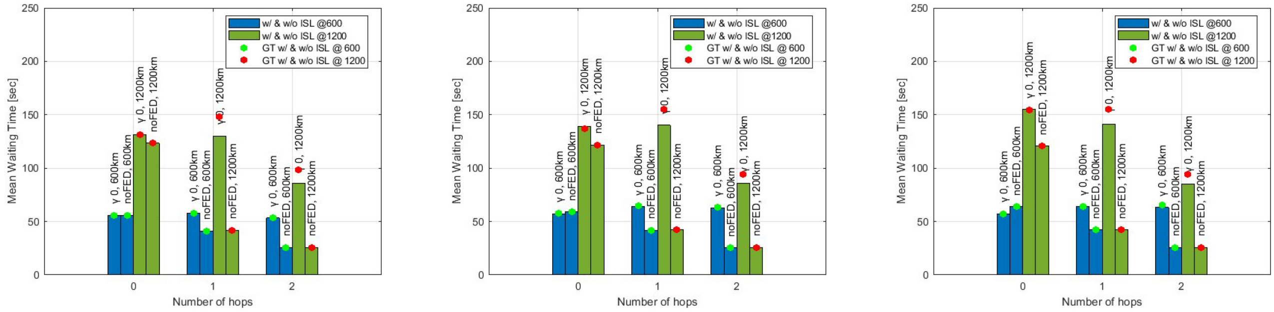
The set of graphs in Figure 5 illustrates that the improvement achieved by our proposed solution in terms of the percentage of completed tasks leads to the side effect of increasing the average mean waiting time for task completion with respect to the policy that does not consider cooperation among SCPs, mainly due to the contribution of the queuing delay. Specifically, the waiting time increases significantly with increasing orbital altitude due to the higher contribution of propagation delay, whereas it shows a decreasing trend with increasing number of hops, as higher resource availability is enabled by the exploitation of ISLs.

Finally, we highlight in the graph in Figure 6 that, even in the scenario more limited in terms of available resources (i.e., 50 satellites per constellation, 1200 km orbital altitude, and one hop neighborhood), the improvement of our proposed solution in terms of percentage of completed tasks remain meaningful also under increasing tasks demand with respect to the no federation policy whose performance remain bounded. Moreover, these results indicate that the analysis conducted in this section is performed under high network load conditions, as the performance begins to decrease starting from a number of requested tasks equal to 3000.

VI. CHALLENGES AND FUTURE WORK

In real-world scenarios, several practical challenges may arise that could affect the performance and reliability of the proposed strategy. While all the following aspects are important, we believe that prioritizing fault tolerance and energy efficiency may address the most immediate barriers to practical deployment. At the same time, we acknowledge that achieving long-term success will also require the integration of robust security mechanisms and adaptive topology management.

- *Link Failures:* Intermittent or persistent ISL failures may hinder task offloading and disrupt coordination within federations. Such failures could reduce the number of reachable federation members, thereby limiting the system's ability to balance load and share resources effectively. In highly dynamic LEO environments, the ability to adapt to failures is fundamental to ensuring uninterrupted service. Mitigating these issues through dynamic re-routing algorithms and redundant communication paths is therefore a top priority for robust system design.
- *CubeSat's power constraints:* CubeSats, typically powered by solar panels and limited-capacity batteries, face significant energy management challenges. Prolonged or inefficient task allocation can quickly drain these limited resources, threatening the continuity of operations. The proposed solution could be enhanced by incorporating satellites' residual energy levels into the cost term of the utility function (see Eq. (4)), enabling energy-aware task distribution. This would help avoid overloading energy-constrained satellites and promote load balancing across the federation.
- *Security Risks:* Potential security threats, such as unauthorized access, data interception, or malicious attacks, pose serious risks to user privacy, data integrity, and service continuity. These concerns are particularly relevant in a multi-tenant, distributed computing scenario such as the one considered in our work. While these



(a) 50 satellites (b) 100 satellites (c) 150 satellites

FIGURE 5. **Mean waiting time for an orbital altitude of 600 km and 1200 km and compromise factor equal to 0. Results are shown for a number of satellites equal to 50 (a), 100 (b), and 150 (c).**

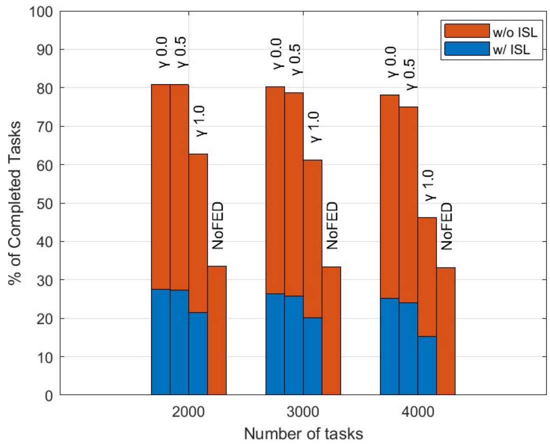


FIGURE 6. **Percentage of completed tasks under increasing task requests for an orbital altitude of 1200 km, a number of satellites equal to 50, and a number of hops equal to 1.**

issues are critical in the broader deployment of satellite networks, addressing them becomes most effective once baseline system resilience and resource management are ensured. Implementing trust-based policies to assess the reliability of satellite nodes could significantly improve the overall security of federated architectures.

- *Dynamic Topology Changes:* While intra-constellation ISLs remain relatively stable, inter-constellation connectivity can change rapidly due to the high mobility of LEO satellites. Managing such topological changes dynamically would be essential for scaling the proposed solution to more heterogeneous and interconnected satellite systems. Although not the focus of this study, dynamic topology management represents a valuable direction for future work.

VII. CONCLUSION

This paper proposes an ISL-aided federation strategy among satellite constellations aimed at realizing an effective computing center in the sky, thus improving task execution while reducing service delay. The designed ISL-aided solution leverages the joint benefits deriving from the implementation of cooperative approaches in a satellite-edge computing multi-tenant scenario and the possibility of utilizing the resources of neighboring satellites reachable via ISLs.

The simulation results prove meaningful benefits that can be achieved by the proposed solution in terms of task execution capability and the flexibility of the ISL-aided federation policy to adapt to the needs of the SCPs.

REFERENCES

[1] J. Wigard et al., "Ubiquitous 6G service through non-terrestrial networks," *IEEE Wireless Commun.*, vol. 30, no. 6, pp. 12–18, Dec. 2023.

[2] A. Rago et al., "Multi-layer NTN architectures toward 6G: The ITA-NTN view," *Comput. Netw.*, vol. 254, Dec. 2024, Art. no. 110725.

[3] G. Araniti, A. Iera, S. Pizzi, and F. Rinaldi, "Toward 6G non-terrestrial networks," *IEEE Netw.*, vol. 36, no. 1, pp. 113–120, 2021.

[4] P. Cassará, A. Gotta, M. Marchese, and F. Patrone, "Orbital edge offloading on mega-LEO satellite constellations for equal access to computing," *IEEE Commun. Mag.*, vol. 60, no. 4, pp. 32–36, Apr. 2022.

[5] G. Araniti, A. Iera, A. Molinaro, S. Pizzi, and F. Rinaldi, "Opportunistic federation of CubeSat constellations: A game-changing paradigm enabling enhanced IoT services in the sky," *IEEE Internet Things J.*, vol. 9, no. 16, pp. 14876–14890, Aug. 2022, doi: 10.1109/JIOT.2021.3115160.

[6] I. Leyva-Mayorga, B. Soret, and P. Popovski, "Inter-plane inter-satellite connectivity in dense LEO constellations," *IEEE Trans. Wireless Commun.*, vol. 20, no. 6, pp. 3430–3443, Jun. 2021.

[7] N. Saeed, A. Elzanaty, H. Almorad, H. Dahrouj, T. Y. Al-Naffouri, and M. S. Alouini, "CubeSat communications: Recent advances and future challenges," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1839–1862, 3rd Quart., 2020.

[8] I. F. Akyildiz and A. Kak, "The Internet of Space Things/CubeSats: A ubiquitous cyber-physical system for the connected world," *Comput. Netw.*, vol. 150, pp. 134–149, Feb. 2019.

[9] S. Xu, X. Wang, and M. Huang, "Software-defined next-generation satellite networks: Architecture, challenges, and solutions," *IEEE Access*, vol. 6, pp. 4027–4041, 2018.

[10] A. Guidotti et al., "Architectures and key technical challenges for 5G systems incorporating satellites," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2624–2639, Mar. 2019.

[11] B. Soret, I. Leyva-Mayorga, S. Cioni, and P. Popovski, "5G satellite networks for Internet of Things: Offloading and backhauling," *Int. J. Satell. Commun. Netw.*, vol. 39, no. 4, pp. 431–444, 2021.

[12] R. Bassoli, F. Granelli, C. Sacchi, S. Bonafini, and F. H. Fitzek, "CubeSat-based 5G cloud radio access networks: A novel paradigm for on-demand anytime/anywhere connectivity," *IEEE Veh. Technol. Mag.*, vol. 15, no. 2, pp. 39–47, Jun. 2020.

[13] A. A. Bisu, A. Gallant, H. Sun, K. Brigham, and A. Purvis, "Telemedicine via Satellite: Improving access to healthcare for remote rural communities in Africa," in *Proc. IEEE Region 10 Humanitarian Technol. Conf. (R10-HTC)*, Malambe, Sri Lanka, 2018, pp. 1–6, doi: 10.1109/R10-HTC.2018.8629855.

[14] L. S. Lawal et al., "Overview of satellite communications and its applications in telemedicine for the underserved in Nigeria: A case study," in *Proc. Int. Conf. Elect. Comput. Commun. Mechatronics Eng. (ICECCME)*, 2022, pp. 1–10, doi: 10.1109/ICECCME55909.2022.9988286.

[15] O. Ben Bahri and L. Kechiche, "Design of nanosatellites constellation to improve the routine of distance consultation," in *Proc. Int. Conf. Women Data Sci.*, 2021, pp. 1–5, doi: 10.1109/WiDSTaif52235.2021.9430212.

[16] D. Hamilton, S. Kohli, P. McBeth, R. Moore, K. Hamilton, and A. Kirkpatrick, "A low earth orbit communication satellites—A positively disruptive technology which could change the delivery of healthcare in rural and Northern Canada," *J. Med. Internet Res.*, vol. 27, Apr. 2025, Art. no. e46113.

[17] Q. Li et al., "Service coverage for satellite edge computing," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 695–705, Jan. 2022.

[18] C. Li, Y. Zhang, R. Xie, X. Hao, and T. Huang, "Integrating edge computing into low earth orbit satellite networks: Architecture and prototype," *IEEE Access*, vol. 9, pp. 39126–39137, 2021.

[19] S. Sthapit, S. Lakshminarayana, L. He, G. Epiphaniou, and C. Maple, "Reinforcement learning for security-aware computation offloading in satellite networks," *IEEE Internet Things J.*, vol. 9, no. 14, pp. 12351–12363, Jul. 2022.

[20] I. Leyva-Mayorga, M. Martinez-Gost, M. Moretti, A. Pérez-Neira, M.Á. Vázquez, P. Popovski, and B. Soret, "Satellite edge computing for real-time and very-high resolution earth observation," *IEEE Trans. Commun.*, vol. 71, no. 10, pp. 6180–6194, Oct. 2023.

[21] J. Sun, X. Chen, Z. Li, J. Wang, and Y. Chen, "Joint optimization of multiple resources for distributed service deployment in satellite edge computing networks," *IEEE Internet Things J.*, vol. 12, no. 3, pp. 2359–2372, Feb. 2025.

[22] Q. Tangand, Z. Fei, B. Li, and Z. Han, "Computation offloading in LEO satellite networks with hybrid cloud and edge computing," *IEEE Internet Things J.*, vol. 8, no. 11, pp. 9164–9176, Jun. 2021.

[23] R. Xie, Q. Tang, Q. Wang, X. Liu, F. R. Yu, and T. Huang, "Satellite-terrestrial integrated edge computing networks: Architecture, challenges, and open issues," *IEEE Network*, vol. 34, no. 3, pp. 224–231, May/Jun. 2020.

[24] X. Cao et al., "Edge-assisted multi-layer offloading optimization of LEO satellite-terrestrial integrated networks," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 2, pp. 381–398, Feb. 2023.

[25] S. S. Hassan, Y. M. Park, Y. K. Tun, W. Saad, Z. Han, and C. S. Hong, "Satellite-based ITS data offloading & computation in 6G networks: A cooperative multi-agent proximal policy optimization DRL with attention approach," *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 4956–4974, May 2024.

[26] J. A. Ruiz-de-Azua, N. Garzaniti, A. Golkar, A. Calveras, and A. Camps, "Towards federated satellite systems and Internet of Satellites: The federation deployment control protocol," *Remote Sens.*, vol. 13, no. 5, p. 982, 2021.

[27] X. Zhu, C. Jiang, L. Kuang, and Z. Zhao, "Cooperative multilayer edge caching in integrated satellite-terrestrial networks," *IEEE Trans. Wireless Commun.*, vol. 21, no. 5, pp. 2924–2937, May 2022.

[28] X. Zhu and C. Jiang, "Delay optimization for cooperative multi-tier computing in integrated satellite-terrestrial networks," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 2, pp. 366–380, Feb. 2023.

[29] Z. Ji, S. Wu, and C. Jiang, "Cooperative multi-agent deep reinforcement learning for computation offloading in digital twin satellite edge networks," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 11, pp. 3414–3429, Nov. 2023.

[30] "Solutions for NR to support non-terrestrial networks (NTN), Release 16," 3GPP, Sophia Antipolis, France, Rep. TR 38.821, Mar. 2023.

[31] Z. Zhang, L. Song, Z. Han, and W. Saad, "Coalitional games with overlapping coalitions for interference management in small cell networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 5, pp. 2659–2669, May 2014.