SFC Enabled Data Delivery for Ultra-Dense LEO Satellite-Terrestrial Integrated Network

Xiaohan Qin, Ting Ma, Zhixuan Tang, Xin Zhang, Xiaoyu Liu and Haibo Zhou
School of Electronic Science and Engineering, Nanjing University, Nanjing, China, 210023.
Email: xhderemail@163.com, majiawan27@163.com, zhixuantang@smail.nju.edu.cn,
zanxin@smail.nju.edu.cn, xyliu0119@163.com, haibozhou@nju.edu.cn

GLOBECOM 2022 - 2022 IEEE Global Communications Conference | 978-1-6654-3540-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/GLOBECOM48099.2022.10001236

***Abstract*—Recently, the rapid-developed mega low earth orbit (LEO) satellite constellation has shown its great potential in cooperating with terrestrial networks to provide seamless global connectivity and high-speed data rate services. However, the heterogeneity of physical resources and diversity of service demands pose challenges for delivering service in an efficient way in the ultra-dense LEO satellite-terrestrial integrated networks (LTIN). When implementing service delivery, service data generally needs a series of on-board processing and then downloading to the terrestrial network for further applications. In this paper, we introduce service function chain (SFC), a sequence of network functions, to process the data on board and propose an efficient multiple service delivery scheme in the LTIN to minimize the total delivery completion time. Considering the heterogeneous resource sharing and competition among multiple SFCs, we formulate the problem as a noncooperative game, which is further proved as a weighted potential game. We design an improved response (IR) algorithm with fast convergence and an adaptive play (AP) algorithm to find the best Nash equilibrium (NE). Extensive simulation results validate the convergence and effectiveness of the proposed algorithms.**

***Index Terms*—Ultra-dense LEO satellite-terrestrial integrated networks, service delivery, SFC embedding, potential game.**

I. Introduction

NOWADAYS, user's increasing demands for seamless connectivity and fast applications are calling for the high capacity and massive access to the terrestrial networks [1], [2]. Fortunately, with the recent rapid development of mega low earth orbit (LEO) satellite constellations [3], [4], the ultra-dense LEO satellite network will play a crucial role in the next generation networks, providing wide coverage and enlarged capacity. With the help of LEO satellites, the ultra-dense LEO satellite-terrestrial integrated networks (LTIN) can serve to guarantee ubiquitous global connectivity as well as handle sophisticated tasks which require high data rate. However, the coordination of heterogeneous physical resources and the establishment of a flexible multiple service delivery framework in the integrated networks turn out to be challenging problems. As a promising networking paradigm, network function virtualization (NFV) [5], [6] emerges to deliver on-demand network services flexibly and achieve efficient resource management. In the NFV-enabled LTIN infrastructure, we introduce service function chain (SFC), a sequence of network functions (NFs) [7] instances (e.g., firewall, load balancer) connected in specific order according to the logic of user's applications. When handling the service requests from LTIN users, the SFC needs to place the corresponding constituent NFs at various network locations, thereby steering data streams to traverse from the source(s) to the destination(s) and ensuring service continuity. Considering the various demands of multiple SFCs and the different execution environments of substrate nodes, a fundamental challenge is how to embed the SFC from each request into the shared physical infrastructure.

Currently, several investigations into SFC embedding have been taken in literatures. With the resource capacity of physical nodes taken into accounts, Cao *et al*. designed an SFC embedding algorithm, which maximized the number of SFC served successfully and minimized the overall deployment time of all SFCs simultaneously [8]. Ren *et al*. investigated the optimal SFC embedding problem for the multicast tasks by a two-stage algorithm [9]. In the first stage, an initial feasible solution was obtained by a Steiner tree, while in the second stage, the solution was optimized by adding new NFs. Pei *et al*. formulated the dynamic SFC embedding problem as a binary integer programming (BIP) model, optimizing the deployment cost for each SFC [10]. However, the above existing works are designed for terrestrial networks. Due to the special characteristics between the LTINs and terrestrial networks, that is, the limited processing ability of LEO satellites and the long distance of inter-satellite links, the existed solutions are inapplicable to the LTINs. Furthermore, owing to the complex and dynamic topology of the ultra-dense LEO satellite networks, the resource competition of multiple SFCs hosting NFs will be more pronounced. Therefore, there is an urgent need to investigate the efficient SFC embedding solution to ensure multiple service delivery in the LTINs.

In this paper, we consider that the service data needs processing by a series of network functions in the satellite network, before being downloaded to the terrestrial network for further application. Via the SFC embedding, a flexible multiple service delivery scheme is proposed in the LTIN, with the heterogeneous resource shared and the competition among multiple SFCs taken into account. To this end, we formulate the problem as a noncooperative game, where each SFC serves as a player who makes decisions independently on the embedding strategy to minimize its service delivery completion latency. Then we design two distributed algorithms for different scenarios to obtain the Nash equilibrium (NE). Our main contributions are listed in the following:

- We formulate the multiple SFC embedding problem as a congestion game adopting the established end-to-end delay calculation method for each flow, which considers the heterogeneous resource sharing and competition among multiple SFCs in the LTIN. Meanwhile, we prove that the game is a weighted potential game with at least one NE.
- We propose two algorithms to achieve the NE of the formulated game. One is the improved response (IR) algorithm with a fast convergence rate, while the other is the adaptive play (AP) algorithm with more possibility to obtain the best solution.
- Simulations are conducted to validate the convergence of the proposed algorithms, which have good performance in minimizing the overall latency of all SFCs and display stability under dynamic network topology.

The remainder of this paper is organized as follows. In Section II, we elaborate the system model and the game formulation. Then we establish the potential game framework in Section III. We next design two algorithms to obtain the NEs in Section IV. Section V analyzes the performance of the proposed algorithms through real-data driven simulations. Finally, Section VI concludes this paper.

II. System Model and Game Formulation

A. Network Model

In the LTIN, one service data is usually generated and collected by a certain satellite, and then processed by multiple NFs hosted on a sequence of LEO satellites, and eventually transmitted to the satellite earth station (SES). As shown in Fig. 1, the SFC for one service request is composed by a sequential concatenation of NFs. By the SFC embedding, service streams traverse these NFs deployed on satellites according to the established order of corresponding SFCs so that data flows can be successfully transferred from the source to the destination, ensuring service delivery.

In the LTIN, the LEO satellites, the SESs and the physical links are separately collected into the sets $\mathcal{S} = \{1, 2, \ldots, S\}$, $\mathcal{V} = \{1, 2, \ldots, V\}$ and $\mathcal{L} = \{1, 2, \ldots, L\}$, where S, V and L represent the number of LEO satellites, SESs and links. For $\forall i, j \in \mathcal{S} \cup \mathcal{V}$, if there is $(i, j) \in \mathcal{L}$, i is the initial node and j is the terminal node. Moreover, the SFCs are denoted by the set $\mathcal{M} = \{1, 2, \ldots, M\}$, where M is the number of SFCs. For the SFC $m \in \mathcal{M}$ with data amount δ_m, its constituent NFs are $F_m = \left\{f_{m1}, f_{m2}, \ldots, f_{m|F_m|}\right\}$. Each of the NFs, which has a one-to-one mapping on the NF types $N_m = \left\{n_{m1}, n_{m2}, \ldots, n_{m|F_m|}\right\}$ can implement a specific task, such as cache, transcoder, firewall, etc., and all NFs are concatenated to achieve the overall service continuity. Here we consider a quasi-static scenario, that is the satellite network topology keeps unchanged during the SFC embedding, but in other times, the network topology may change with the movement of satellites.

B. Communication Model

The i-th SFC is served as player i with its embedding decision $\overline{a}_i \in A_i$, where A_i denotes the strategy set ($i = 1, \ldots, M$). To be specific, $\overline{a}_i = \left\{a_{i0}, a_{i1}, a_{i2}, \ldots, a_{i|F_i|}, a_{i,|F_i|+1}\right\}$, where $a_{i0} \in \mathcal{S}$ is the source LEO satellite, $a_{ij} \in \mathcal{S}$ is the processing satellite specified by player i to embed f_{ij} $(j = 1, \ldots, |F_i| - 1)$ and $a_{i,|F_i|+1} \in \mathcal{V}$ is the destination SES. For player $i \in \mathcal{M}$ and its strategy $\overline{a}_i \in A_i$, the end-to-end latency served by the system includes three parts, and we introduce them as follows.

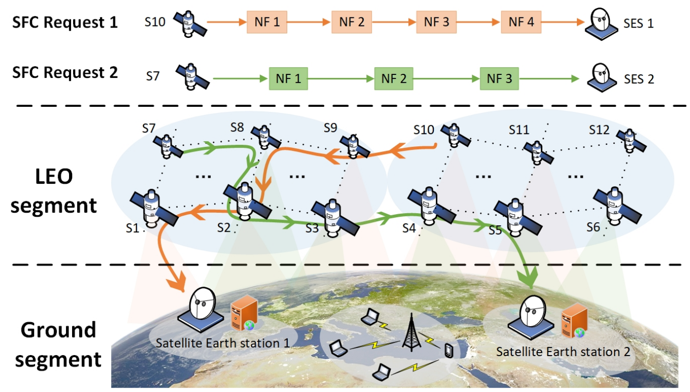

Fig. 1. SFC-based service provision for TLIN

1) *Processing Time*: In the system, each LEO satellite is capable of hosting multiple NFs from different SFCs, but serving more SFC leads to more processing time. Here ρ_s represents the time caused by LEO satellite $s \in \mathcal{S}$ to process one data bit. Accordingly, the processing time for player i on satellite s is given by

$$P_s = \rho_s \delta_i \left|\Upsilon_s\right|, \tag{1}$$

where $|\Upsilon_s|$ represents the number of players who deploy their NFs on satellite s.

2) *Transmission Time*: Let B_l represent the bandwidth of link $l = (t, r)$, which is divided equally by each player flowing through t. And the transmission rate for player i along l is defined as

$$R_l = \frac{B_l}{|\Psi_l|}, \tag{2}$$

where $|\Psi_l|$ represents the number of players transmitting their data through the link l. Then the transmission time of player i along l can be calculated as

$$\eta_l = \frac{\delta_i}{R_l}. \tag{3}$$

3) *Propagation Time*: We define ϵ_l as the distance of the link l and the propagation time of player i through link l is

$$\epsilon\left(l\right) = \frac{\epsilon_l}{v}, \tag{4}$$

where v represents the signal propagation rate of links.

We define $a_{-i} = \{\overline{a}_1, \overline{a}_2, \ldots, \overline{a}_{i-1}, \overline{a}_{i+1}, \ldots, \overline{a}_M\}$ as the embedding strategy set of all players except the i-th one. Considering that multiple NFs from different players can be hosted on the same LEO satellite, we denote $\Upsilon_s\left(\overline{a}_i, a_{-i}\right)$ as the set of players who embed their NFs on the satellite s. That is, $\Upsilon_s(\overline{a}_i, a_{-i}) = \left\{i' \in \mathcal{M} : \exists j \in 1, \ldots, |F_{i'}|, a_{i'j} = s, a_{i'j} \in \overline{a}_{i'}, \overline{a}_{i'} \in (\overline{a}_i, a_{-i})\right\}$. Moreover, when players select two identical nodes twice in succession, it means that they will transmit data over the same link. Similarly, we define $\Psi_l(\overline{a}_i, a_{-i})$ as the set of players who traverse the same link l. That is, $\Psi_l\left(\bar{a}_i, a_{-i}\right) = \left\{i' \in \mathcal{M} : \exists j \in 0, \ldots, |F_{i'}|, (a_{i'j}, a_{i'j+1}) = l, a_{i'j} \in \overline{a}_{i'}, \overline{a}_{i'} \in (\overline{a}_i, a_{-i})\right\}$

According to Eq. (1), we calculate the total processing time experienced by player i as

$$d_i^{(proc)}(\overline{a}_i, a_{-i}) = \sum_{s \in \overline{a}_i} \rho_s \delta_i |\Upsilon_s(\overline{a}_i, a_{-i})|. \tag{5}$$

Based on Eq. (2) and (3), for player $i \in \mathcal{M}$ and its strategy $\overline{a}_i \in A_i$, its total link transmission time is computed as

$$d_i^{(tran)}(\overline{a}_i, a_{-i}) = \sum_{j=0}^{|F_i|} \sum_{l \in (a_{ij}, a_{i,j+1})} \frac{\delta_i}{R_l(\overline{a}_i, a_{-i})}. \tag{6}$$

According to Eq. (4), the total propagation time required by the player i is given as

$$d_i^{(prop)}(\overline{a}_i) = \sum_{j=0}^{|F_i|} \sum_{l \in (a_{ij}, a_{i,j+1})} \frac{\epsilon_l}{v}. \tag{7}$$

To sum up, the total delivery completion time of an embedded SFC is the sum of the total link transmission time, the total processing time and the total propagation time. Let $D_i(\overline{a}_i, a_{-i})$ represent the total latency of player i under the strategy set $(\overline{a}_i, a_{-i})$, that is

$$D_i(\overline{a}_i, a_{-i}) = d_i^{(tran)}(\overline{a}_i, a_{-i}) + d_i^{(proc)}(\overline{a}_i, a_{-i}) + d_i^{(prop)}(\overline{a}_i). \tag{8}$$

C. Game Formulation

According to the above established communication model, the latency experienced by each player depends not only on its own embedding strategy, but also on other players' due to the shared physical infrastructure. Specifically, when there are too many identical elements in their strategy (for example, choosing the same LEO to host NFs or using the same wireless channel, etc.), the players may experience higher overall delay (including processing or transmission time). Considering the resource competition and dependent relationship among players, game theory is a powerful tool to analyze the decision-making process of multiple players. In this paper, we consider that each player is rational, aiming at minimizing its delivery completion time by strategy selection, i.e.,

$$\min_{\overline{a}_i \in A_i} \quad D_i(\overline{a}_i, a_{-i}) \quad \forall i \in \mathcal{M}, \tag{9}$$

where $D_i(\overline{a}_i, a_{-i})$ satisfies Eq. (8).

Accordingly, the above problem in the LTIN can be formulated as a congestion game $\mathcal{G} = \left\{\mathcal{M}, \mathcal{S}, \{A_i\}_{i \in \mathcal{M}}, \{D_i\}_{i \in \mathcal{M}}\right\}$, referred to as the multiple SFC embedding game (MSEG). In the game $\mathcal{G}$, $\mathcal{M}$ represents the player set, $\mathcal{S}$ is the LEO satellite set, A_i is the strategy set of player i and D_i is the total delivery completion time experienced by player i. In the game theory, NE is the most important concept to analyze the outcome of the strategic interaction of multiple decision-makers, which also corresponds to a stable state of the network. We give further analysis in the next section.

III. Analysis of Nash Equilibrium

In this section, we first introduce the definitions of NE and weighted potential game. Then, the existence of NE in the formulated game $\mathcal{G}$ is proved.

Definition 1 (Nash Equilibrium, NE): A strategy profile $(\overline{a}_1^*, \overline{a}_2^*, \ldots, \overline{a}_M^*)$ is a Nash Equilibrium if and only if no player can decrease its latency function by deviating unilaterally, i.e.,

$$D_i\left(\overline{a}_i^*, a_{-i}^*\right) \le D_i\left(\overline{a}_i, a_{-i}^*\right), \quad \forall i \in \mathcal{M}, \ \forall \overline{a}_i \in A_i. \tag{10}$$

The NE can be interpreted as a self-stability state of the system. In this case, all players can reach a mutually satisfactory condition where no one has the intention to change its strategy because its latency function cannot be unilaterally reduced by its own changes.

Definition 2 (Weighted Potential Game): A game is called a weighted potential game if there is a function Φ such that for $\forall i \in \mathcal{M}$, $\forall \overline{a}_i, \overline{a}_i' \in A_i$, it satisfies

$$D_i(\overline{a}_i, a_{-i}) - D_i\left(\overline{a}_i', a_{-i}\right) = \omega_i\left(\Phi(\overline{a}_i, a_{-i}) - \Phi\left(\overline{a}_i', a_{-i}\right)\right), \tag{11}$$

where Φ is called a weighted potential function.

In a weighted potential game, the deviation in the cost of an arbitrary player is proportional to the change of the potential function for the whole system. According to [11], the weighted potential game has the finite improvement property (FIP) that after a finite number of iterations, the system can always achieve a pure strategy NE. In Theorem 1, we will construct a potential function to prove that the MSEG is a weighted potential game.

Theorem 1: The multiple SFC embedding game (MSEG) $\mathcal{G}$ is a weighted potential game with at least one NE.

Proof: We design a potential function of the congestion game as

$$\Phi(\overline{a}_i, a_{-i}) = \sum_{i \in \mathcal{M}} \frac{d_i^{(prop)}(\overline{a}_i)}{\delta_i} + \sum_{l \in \mathcal{L}} \sum_{k=1}^{|\Psi_l(\overline{a}_i, a_{-i})|} \frac{k}{B_l} + \sum_{s \in \mathcal{S}} \sum_{k=1}^{|\Upsilon_s(\overline{a}_i, a_{-i})|} \rho_s k. \tag{12}$$

Assume that player i changes its strategy from $\overline{x}$ to $\overline{y}$ while others keep their strategies unchanged. Then we can compute the change of the potential function Φ between $\overline{x}$ and $\overline{y}$ as $\Phi(\overline{x}, a_{-i}) - \Phi(\overline{y}, a_{-i})$ based on Eq. (12). From Eq. (8), the difference between the delay function for player i is denoted as $D_i(\overline{x}, a_{-i}) - D_i(\overline{y}, a_{-i})$.

We can prove that $D_i(\overline{a}_i, a_{-i}) - D_i\left(\overline{a}_i', a_{-i}\right) = \delta_i\left(\Phi(\overline{a}_i, a_{-i}) - \Phi\left(\overline{a}_i', a_{-i}\right)\right)$. Therefore, $\mathcal{G}$ is a weighted potential game with weight $\omega = \{\delta_i\}_{i \in \mathcal{M}}$. The detailed proof is similar to the Theorem 3.1 in [12] and we omit the proof due to page constraints. ■

IV. Algorithm Design

In this section, we present two distributed algorithms to obtain the NE of the formulated game. First, we introduce the IR algorithm which can obtain an NE quickly, but may lead to a locally optimal solution. Then, to improve the optimality, AP algorithm is designed to get the optimal NE of MSEG.

Algorithm 1 Improved Response (IR) Algorithm

Input: A LTIN $N = (\mathcal{S}, \mathcal{V}, \mathcal{L})$ and M SFCs with $SFC_m = (F_m, \delta_m, \overline{a}_m, A_m), m \in \mathcal{M}$.
Output: NE solutions $(\overline{a}_1^*, \overline{a}_2^*, \ldots, \overline{a}_M^*)$.

Initialize the iterative round $r \leftarrow 1$;
for i in $\mathcal{M}$ **do**
  Randomly select an embedding decision from the whole strategy set as the initial decision $\overline{a}_i(1) \in A_i$;
  Execute the initial decision $\overline{a}_i(1)$;
end for
while *StopConditionNotMet()* **do**
  for i in $\mathcal{M}$ **do**
    Update the iterative round $r \leftarrow r + 1$;
    Keep a_{-i} unchanged, i.e. $a_{-i}(r) = a_{-i}(r-1)$;
    Traverse the whole possible decisions in the strategy set A_i to find the improved response decision $\overline{a}_i^*$, which satisfies $\overline{a}_i^* = arg\min\limits_{\overline{a}_i} D_i(\overline{a}_i, a_{-i}(r))$;
    Set $\overline{a}_i(r) = \overline{a}_i^*$;
    Execute the strategy $\overline{a}_i(r)$;
  end for
end while

A. Improved response algorithm

According to Theorem 1, MSEG is a weighted potential game, so we can obtain an NE by only allowing one player to select an improved strategy in each iterative round. Specifically, one player updates its strategy to minimize its delay function in each iteration, while others keep their strategies unchanged. Once the stop condition is met (e.g., all players have updated their strategies, or the maximum iterative round allowed has been exceeded), the algorithm will terminate. The details of the IR algorithm are given in Algorithm 1.

With the FIP, IR algorithm can achieve an NE of the MSEG after finite iterative rounds, but the solution cannot be guaranteed to the optimum, that is the potential function may not be minimized. To find the best NE of the MSEG, we further propose the adaptive play (AP) algorithm.

B. Adaptive Play (AP) algorithm

Let $\mathbf{p}_i(r) = \left\{p_{i1}(r), \ldots, p_{i|A_i|}(r)\right\}$ denote the strategy selection probability vector of player i. For each iteration, there is $\sum_{j=1}^{|A_i|} p_{ij}(r) = 1$, where $p_{ij}(r)$ is the probability that player i chooses the j-th embedding decision in the r-th round. The exploration factor β is used to jump out of the locally optimal solutions.

In AP algorithm, each player $i \in \mathcal{M}$ initially selects a strategy from its strategy set A_i with evenly distributed probability $p_{i,j}(r=1) = \frac{1}{|A_i|}, (j = 1, \ldots, |A_i|)$. Then in each iteration, player i updates its decision from $\overline{a}_i(r)$ to $\overline{a}_i(r+1)$, which is chosen from A_i according to the updated strategy selection probability vector $\mathbf{p}_i(r+1)$ as follows:

$$p_{ij}(r) = \frac{exp\left(-\frac{\beta}{\delta_i} D_i(\overline{a}_{ij}, a_{-i}(r))\right)}{\sum_{\overline{a}_{ik} \in A_i} exp\left(-\frac{\beta}{\delta_i} D_i(\overline{a}_{ik}, a_{-i}(r))\right)}, \tag{13}$$

where $\overline{a}_{ij}$ denotes the j-th strategy in A_i, and β denotes the exploration factor. The larger β is, the higher the probability that players select the improved strategy (corresponding to the solutions of the IR algorithm) is, while the smaller β is, the slower the convergence rate is. When the stop condition is met, the algorithm will terminate. The AP algorithm is summarized in Algorithm 2, and its convergence is proved in Theorem 2.

Theorem 2: The adaptive play (AP) algorithm can converge to the stationary distribution as follows:

$$\pi(\mathbf{a}) = \frac{exp(-\beta\Phi(\mathbf{a}))}{\sum_{\mathbf{a}' \in \mathbf{A}} exp(-\beta\Phi(\mathbf{a}'))}. \tag{14}$$

Proof: The analysis on the AP algorithm in convergence and optimality is following similar proofs of Theorem 3 and Theorem 4 in [13]. We omit the proof due to page limitation. ■

Algorithm 2 Adaptive Play (AP) Algorithm

Input: A LTIN $N = (\mathcal{S}, \mathcal{V}, \mathcal{L})$ and M SFCs with $SFC_m = (F_m, \delta_m, \overline{a}_m, A_m), m \in \mathcal{M}$.
Output: NE solutions $(\overline{a}_1^*, \overline{a}_2^*, \ldots, \overline{a}_M^*)$.

Initialize the iterative round $r \leftarrow 1$, the probability vector $p_{i,j}(r=1) = 1/|A_i|, \forall i \in \mathcal{M}, \forall j = 1, \ldots, |A_i|$ and the exploration factor β;
for i in $\mathcal{M}$ **do**
  Select an initial decision $\overline{a}_i(1) \in A_i$ from the strategy set based on the probability vector $p_{i,j}(1)$;
  Execute the initial decision $\overline{a}_i(1)$;
end for
while *StopConditionNotMet()* **do**
  for i in $\mathcal{M}$ **do**
    Update the iterative round $r \leftarrow r + 1$;
    Keep a_{-i} unchanged, i.e. $a_{-i}(r) = a_{-i}(r-1)$;
    Traverse all the possible strategies of player i to calculate $D_i\left(\overline{a}_{ij}, a_{-i}(r)\right), \forall j = 1, \ldots, |A_i|$;
    Update the strategy selection probability vector according to Eq. (13);
    Select the new strategy $\overline{a}_i(r)$ according to the updated probability vector $p_{ij}(r)$;
    Execute the strategy $\overline{a}_i(r)$;
  end for
end while

V. Performance Evaluation

In this section, we conduct simulations to evaluate the convergence and effectiveness of the proposed algorithms. In the ultra-dense LEO satellite segment, the distance information and topology relation among tens of thousands of LEO satellites is obtained by learning from the Starlink constellation design [14], [15]. For the terrestrial-satellite links (TSL), SESs within satellite coverage can establish line-of-sight (LoS) links with the satellites, and the position information of SESs is obtained according to the real location data deployed on the Earth. Based on this, we choose a specific region on the ground as an example to analyze our multiple service delivery solutions in the LTIN. Table I summarizes the detailed parameter settings.

TABLE I
SIMULATION PARAMETERS

Parameter	Value
Bandwidth of ISLs	[100 , 150] Mbps
Bandwidth of TSLs	[150 , 200] Mbps
Data amount of each SFC δ_i	[100 , 200] KB
NF processing time of LEO satellite ρ_s	[5 , 10]$\times 10^{-4}$ ms
Maximum elevation angle of LEO satellite	45^o

A. Convergence Analysis

Here we use the greedy search (GS) algorithm as a benchmark to obtain the global optimum. Considering the complexity of the GS algorithms, we first select a small network instance for performance comparison, i.e., a ground region from ($46^o02^{'}$S , $57^o90^{'}$W) to ($63^o32^{'}$S , $71^o00^{'}$W). There are 8 SFCs, i.e., $M = 8$, and the number of NFs for each SFC is the minimum hop determined by the Dijkstra algorithm. We set the maximum iterative round as 100. For the AP algorithm, the exploration factor β is chosen from [2, 4, 6]. Fig. 2 shows the convergence behavior of the IR and AP algorithms. After a few iterations, the IR algorithm can converge to the globally optimal value reached by the GS algorithm, which indicates the effectiveness of the IR algorithm in such a small network scenario. For the AP, as the iterative round increases, the value of potential function is reduced to near the global optimum and converged at $r = 20$ with $\beta = 6$, but has not converged within maximum iterative rounds with $\beta = 2$ and $\beta = 4$. To achieve its convergence, more rounds are needed, as we explain later.

B. Performance Analysis

We compare the performance of the AP and IR algorithms in finding the optimal NE. Here we consider the specific region from ($22^o26^{'}$S , $144^o96^{'}$W) to ($77^o81^{'}$S , $167^o81^{'}$W), and the SFC number is 65, i.e., $M = 65$. In later simulations, we will follow these parameter settings, and not explain repeatedly. The exploration factor of the AP is chosen from [2, 5, 8]. Fig. 3 depicts the algorithm performance in terms of minimizing the overall delivery completion time of all SFCs. Because the MSEG is a weighted potential game, the change of the total latency of SFCs can reflect the deviation in the potential function value, as demonstrated in Theorem 1. Notice that the IR takes about 80 rounds to reach an NE, while it takes about 110 rounds for the AP with $\beta = 8$. Notably, a larger exploration factor leads to a faster convergence. However, it is noted that the overall latency obtained by the AP is smaller than that in the IR, which indicates that the AP algorithm has the greater ability to find the optimal NE of the MSEG.

C. Topology Varying Adaption

In real scenarios, LEO satellite moves along its orbit periodically, so the network topology is time-varying. Thus, we simulate to observe the performance of the proposed algorithms under the changing networks. For convenience, we take the IR as an example, as the workflow of the AP is similar to the IR. We first let the system reaches an NE by the IR as we have done in previous simulations. As shown in Fig. 4, after 150 rounds, the system achieves a stable state, i.e., NE denoted by the most left-hand-side green square. Then at the 200-th round, all satellites are allowed to move at the maximum speed of 4km/s for 5 seconds in any direction, and the bandwidths of ISLs and TSLs are redistributed at the same time. At this point, the total latency changes abruptly represented by the most left-hand-side blue star. This is because the link bandwidth changes at this moment, leading to the change of the latency experienced by the service data along the path corresponding to the NE before, but it is no longer the locally optimal of the system after the change. Therefore, the system will re-iterate to reach a new NE. After 65 rounds, at $r = 265$, the system reaches a new NE again. Similarly, we observe another topology changing. It can be seen that the system cost fluctuates within a narrow range each time and the proposed algorithm can reach a new NE quickly. Therefore, our algorithm can quickly adapt to the topology changing in the practical scenario with a relatively small system overhead.

D. Impact of Simulation Parameters

In order to further evaluate the proposed algorithms, we compare their performance under different system parameters. Here the random search (RS) algorithm is used for comparison, where each SFC randomly selects a path from its strategy set in each iteration. Due to the similar performance between the IR and AP algorithm according to Fig. 3, here we show the solution of the AP algorithm only. Fig. 5 shows the impact of SFC amount on the overall service delivery latency, where the network topology and parameters are fixed. Intuitively, the total completion latency of all SFCs increases when the number of SFCs increases. Meanwhile, the competition among more SFCs becomes more obvious, so our game-theoretical method can coordinate resource sharing among SFCs and fulfill better performance in overall latency than the RS algorithm.

VI. CONCLUSION

In this paper, we have proposed an efficient multiple service delivery scheme in order to minimize the total delivery completion time, thereby coordinating heterogeneous physical resources and ensuring service continuity. Considering the resource sharing and competition among different SFCs, we have formulated the problem as a congestion game. Based on the potential game theory, two algorithms have been proposed for different application scenarios. Simulation results have shown that our algorithms perform well in overall delivery latency and display stability under varying network topology. In the future work, we intend to study the influence of nodes capacity constraints on the formulated game.

ACKNOWLEDGMENT

This work was supported in part by the National Key R&D Program of China under Grant 2020YFB1806104, in part by Natural Science Foundation of Jiangsu Province Youth Project

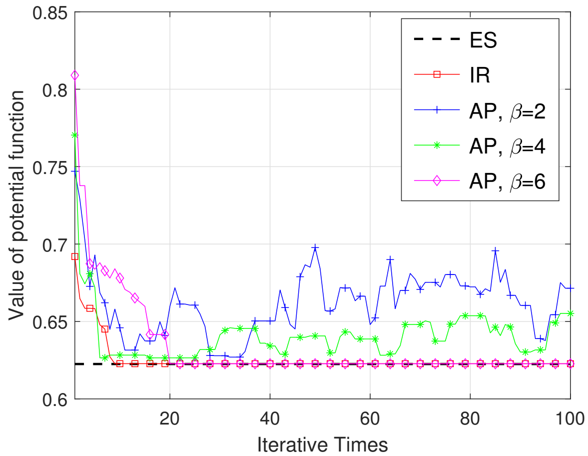

Fig. 2. Convergence behavior of algorithms.

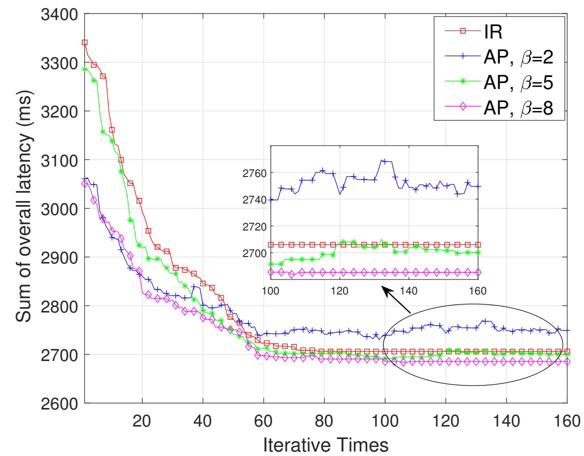

Fig. 3. Sum of overall latency of the IR and AP algorithm.

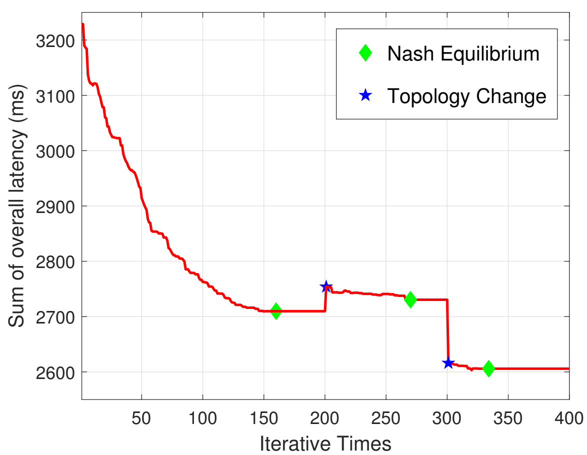

Fig. 4. Topology varying adaption of the IR algorithm.

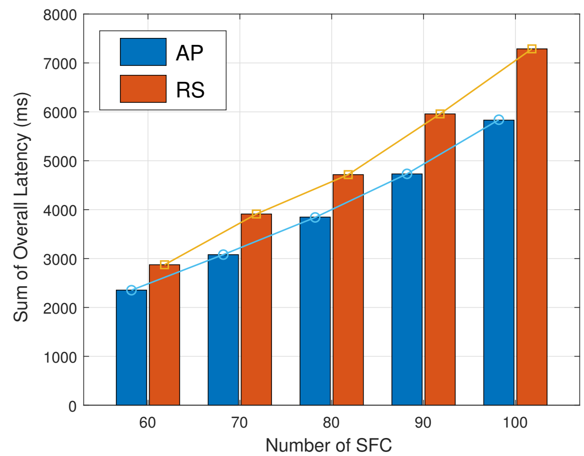

Fig. 5. Impact of SFC amount.

(BK20180329) and in part by Innovation and Entrepreneurship of Jiangsu Province High-level Talent Program.

References

[1] J. Liu, Y. Shi, Z. M. Fadlullah, and N. Kato, "Space-air-ground integrated network: A survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2714–2741, 2018.

[2] T. Ma, H. Zhou, B. Qian, N. Cheng, X. Shen, X. Chen, and B. Bai, "UAV-LEO Integrated Backbone: A Ubiquitous Data Collection Approach for B5G Internet of Remote Things Networks," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 11, pp. 3491–3505, 2021.

[3] T. Butash, P. Garland, and B. Evans, "Non-geostationary satellite orbit communications satellite constellations history," 2021.

[4] F. SpaceX, "update. 2018. SPACEX NON-GEOSTATIONARY SATELLITE SYSTEM."

[5] K. Kaur, V. Mangat, and K. Kumar, "Architectural Framework, Research Issues and Challenges of Network Function Virtualization," in *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)*. IEEE, 2020, pp. 474–478.

[6] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Communications surveys & tutorials*, vol. 18, no. 1, pp. 236–262, 2015.

[7] H. Hantouti, N. Benamar, and T. Taleb, "Service Function Chaining in 5G & Beyond Networks: Challenges and Open Research Issues," *IEEE Network*, vol. 34, no. 4, pp. 320–327, 2020.

[8] H. Cao, J. Du, H. Zhao, D. X. Luo, N. Kumar, L. Yang, and F. R. Yu, "Resource-Ability Assisted Service Function Chain Embedding and Scheduling for 6G Networks With Virtualization," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 4, pp. 3846–3859, 2021.

[9] B. Ren, D. Guo, Y. Shen, G. Tang, and X. Lin, "Embedding service function tree with minimum cost for NFV-enabled multicast," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 5, pp. 1085–1097, 2019.

[10] J. Pei, P. Hong, K. Xue, and D. Li, "Efficiently embedding service function chains with dynamic virtual network function placement in geo-distributed cloud system," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 10, pp. 2179–2192, 2018.

[11] D. Monderer and L. S. Shapley, "Potential games," *Games and economic behavior*, vol. 14, no. 1, pp. 124–143, 1996.

[12] J. Li, W. Shi, Q. Ye, N. Zhang, W. Zhuang, and X. Shen, "Multiservice function chain embedding with delay guarantee: A game-theoretical approach," *IEEE Internet of Things Journal*, vol. 8, no. 14, pp. 11 219–11 232, 2021.

[13] Y. Xu, J. Wang, Q. Wu, A. Anpalagan, and Y.-D. Yao, "Opportunistic spectrum access in cognitive radio networks: Global optimization using local interaction games," *IEEE Journal of Selected Topics in Signal Processing*, vol. 6, no. 2, pp. 180–194, 2011.

[14] Q. Chen, G. Giambene, L. Yang, C. Fan, and X. Chen, "Analysis of inter-satellite link paths for LEO mega-constellation networks," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 3, pp. 2743–2755, 2021.

[15] N. Cheng, W. Quan, W. Shi, H. Wu, Q. Ye, H. Zhou, W. Zhuang, X. Shen, and B. Bai, "A comprehensive simulation platform for space-air-ground integrated network," *IEEE Wireless Communications*, vol. 27, no. 1, pp. 178–185, 2020.