Efficient Algorithms for Service Chaining in NFV-Enabled Satellite Edge Networks

Qiufen Xia, *Member, IEEE*, Guijie Wang, Zichuan Xu, *Member, IEEE*, Weifa Liang, *Senior Member, IEEE*, and Zhou Xu

***Abstract*—Satellite-terrestrial networks are emerging as the next-generation networking paradigm for Beyond-5 G (B5G) and 6 G networks. Meanwhile, Mobile Edge Computing (MEC) is envisioned as the key technology to provide network services within the proximity of users, by deploying computing resource in ground locations that are close to users. With the fast deployment of Low-Earth-Oribt (LEO) satellites, a new paradigm of MEC is emerging by enabling LEO satellites serving as edge servers in lower orbits that are close to ground users. In this way, the ground users can be further served by LEO satellites in lower orbits instead of conventional high-orbit satellites. Also, since LEO satellites provide shorter paths from users to services, the performance is enhanced compared with ground MEC networks. In this paper, we aim to enable low-latency network services in a Satellite Edge Computing (SEC) network that integrates the MEC and satellite-terrestrial networks. In particular, we consider that each network service is composed of a sequence of Virtualized Network Functions (VNFs), where the traffic of user requests has to be processed by the VNFs in a service chain in the specified order before reaching its destination. To this end, we first formulate a delay-aware service chaining problem in an SEC network to minimize the average delay of implementing a user request, by jointly placing VNFs to LEO satellites in the SEC network and routing the traffic of each user request from its source to destination. We then devise an approximation algorithm with an approximation ratio for the problem in an SEC network with a single user request, by devising a novel concept of *chaining orbit* and auxiliary graph construction technique. We also design an online algorithm for the online delay-aware service chaining problem in an SEC network, if user requests arrive into the system without the knowledge of their arrivals and the network delays are uncertain. We finally evaluate the performance of the proposed algorithms using real satellite network topologies, and results show that the proposed algorithms achieve 28.5% lower delay than their counterparts.**

***Index Terms*—Approximation and online algorithms, mobile edge computing, network function chaining, satellite edge network.**

I. Introduction

With the development of the Internet of things (IoT), more and more IoT devices are deployed in remote areas for environmental monitoring [35], disaster early warning [41], ocean transportation [53] and so on. With these overwhelming demands from IoT devices, terrestrial networks alone cannot provide the huge traffic solutions in a seamless, high-rate, and reliable manner [7], [38], [57]. The technique of satellite-terrestrial network is emerging as a promising solution to meet the demands of IoT devices in remote areas, where satellites in the space and base stations in the ground collaboratively provide services for various IoT applications. For example, in the case of natural disasters with damaged ground network infrastructure, the images that capture the natural disasters can be transmitted through the satellite network in the space [3]. Also, satellite-terrestrial networks can provide ubiquitous coverage for remote areas such as mountain areas or rural areas that lack the coverage of ground base stations [53]. However, conventional satellite-terrestrial networks may not meet the computing requirements of IoT services, as they are mainly designed for communication purposes. Recently, Mobile Edge Computing (MEC) [9], [28] deploys edge servers within close locations of users to offer low-latency computing services. Instead of deploying edge servers in the ground, satellites with computing resource can serve the IoT devices when they are right above the remote area, due to the fast development of Low Earth Orbit (LEO) satellites. For instance, many satellite constellations such as Iridium II [12], SpaceX Starlink [16] and OneWeb [27] have been built or completed to serve ground users in remote areas. As such, the integration of the low-latency computing and ubiquitous coverage abilities of MEC and satellite-terrestrial networks is considered as the next trend of B5G or 6 G networks, which allows Low-Earth-Orbit (LEO) satellites serving as edge servers that are located in close orbits of a satellite-terrestrial network.

IoT applications in a satellite edge computing (SEC) network usually need to transmit data traffic to remote data centers via LEO applications for processing. To guarantee the security of such data transfer and processing, various network functions, such as firewalls, Intrusion Detection Systems (IDSs), etc., are required. Different sequences of such network functions

Manuscript received 10 August 2022; revised 12 May 2023; accepted 28 August 2023. Date of publication 6 September 2023; date of current version 4 April 2024. The work of Qiufen Xia and Zichuan Xu was supported by the National Natural Science Foundation of China (NSFC) under Grants 62172071, 62172068, in part by National Science Foundation of Liaoning with Grant 2023-MS-111, in part by "Xinghai scholar" program, LiaoNing Revitalization Talents Program under Grant 2022RG04. The work of Weifa Liang was supported by the City University of Hong Kong under Grant 9380137/CS. Recommended for acceptance by A. Shami. *(Corresponding author: Zichuan Xu.)*

Qiufen Xia and Guijie Wang are with the International School of Information Science and Engineering, Dalian University of Technology, Dalian, Liaoning 116024, China, and also with the Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, Dalian, Liaoning 116024, China (e-mail: qiufenxia@dlut.edu.cn; wgj1128@mail.dlut.edu.cn).

Zichuan Xu and Zhou Xu are with the School of Software, Dalian University of Technology, Dalian, Liaoning 116024, China, and also with the Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, Dalian, Liaoning 116024, China (e-mail: z.xu@dlut.edu.cn; 21917040@mail.dlut.edu.cn).

Weifa Liang is with the Department of Computer Science, City University of Hong Kong, Hong Kong (e-mail: weifa.liang@cityu.edu.hk).

Digital Object Identifier 10.1109/TMC.2023.3312352

1536-1233 © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 13,2024 at 13:29:56 UTC from IEEE Xplore. Restrictions apply.

form network services to process the data of IoT applications in the SEC network before reaching their destinations. However, conventional hardware-based network functions are not suitable for deployment in LEO satellites, as they increase their payloads and the maintenance costs. For instance, if a hardware network function of an LEO satellite malfunctions, the whole satellite may not be able to provide any services, thereby wasting its launching cost. Recently, Network Function Virtualization (NFV) is considered to be a promising solution to provide flexible and low-cost services, by transferring network functions from dedicated hardware to virtual machines (VMs) that run on commercial hardware to reduce dependence on underlying hardware [54], [55], thereby increasing the functionalities of satellites and saving launching costs significantly [36], [63].

In this paper, we aim to enable low-latency service chaining for ground users in remote areas in a Satellite Edge Computing (SEC) network. Specifically, we consider enabling low-latency and secure data transmissions and processing for ground users in remote areas that are out-of-reach of ground base stations. Enabling low-latency service chaining in an SEC network however poses many fundamental challenges. First, each LEO satellite usually has limited computing resource, and the availability of such computing resource is dynamic. Also, each LEO satellite circulates the Earth and the length of each circulation is short (90 to 120 minutes [4]). This makes the availability of each LEO satellite to ground users intermittent. As such, placing network functions to LEO satellites with enough resources becomes more difficult, compared with ground networks. Second, LEO satellites usually are distributed into orbits at different heights, forming a *hierarchical SEC network*. This means that the topology of the inter-satellite network is dynamically changing as the time goes, which further complicates the service chaining of user requests. Simply placing network functions without considering such dynamic inter-satellite communications may interrupt the data transmissions among network functions of a service chain. Third, the transmission delays of links and processing delays of LEO satellites depend on various factors, such as congestion levels of links, workload of LEO satellites, and locations of LEO satellites. For example, the wireless transmission delays from ground users to an LEO satellite jointly depend on the distance and channel status among them. Optimization of the service chaining and data transmission under such uncertainty is fundamentally challenging. To address these challenges, in this paper we investigate the *delay-aware service chaining problem in an SEC network* with the aim to minimize the delay of a user request that performs data transmissions and processing in a service chain.

To the best of our knowledge, we are the first to consider the delay-aware service chaining in a hierarchical SEC network with LEO satellites being distributed into orbits with different heights. Although there are studies focusing on satellite-terrestrial networks, most of them ignored the merging of MEC and LEO satellites [21], [50], [58]. The research of satellite edge computing is in its very early stage. Although there are several studies that aim to enable the merge of MEC and satellite networks, most of them did not investigate the service chaining problem [6], [17], [51] and assumed that network delays are given [14], [40], [45], [63].

The main contributions of this paper are as follows.

- We formulate the delay-aware service chaining problem in an SEC network, with the objective of minimizing the average delay of each NFV-enabled request, subject to the capacity constraints of LEO satellites and the budget constraints of each NFV-enabled request.
- We devise an approximation algorithm with an approximation ratio for the delay-aware service chaining problem in an SEC network with a single NFV-enabled request.
- We design an efficient online algorithm for the online delay-aware service chaining problem in an SEC network with multiple NFV-enabled requests arriving dynamically and with uncertain delays.
- We evaluate the performance of the proposed algorithms by extensive simulations and the results show that the performance of the proposed algorithms outperforms its counterparts by at around 28.5% less average delay.

The organization of this paper is as follows. Section II introduces the state-of-the-art on this study. Section III describes the system model and problem formulation of the delay-aware service chaining problem in an SEC network. Section IV devises an approximation algorithm for the problem with a single service chaining request. Section V designs an efficient algorithm for the online version of the problem with uncertain delays and multiple requests arriving into the system dynamically.

II. Related Work

Recently, satellite-terrestrial networks promise to provision high-performance network services for ground users [24], [43], [47], [49]. The technique of LEO satellites is one of the primary focus of these existing studies due to the low latency, low path loss, and low production and launching cost, as important supplement of terrestrial networks [46]. Most of these studies focus on service placement and routing or request dispatching [34], [48]. For example, Varasteh et al. [48] proposed a cost-effective scheme for joint service placement and routing to guarantee service requirements in dynamic and heterogeneous space-air-ground integrated networks. Li et al. [34] studied the problem of joint request dispatching and service placement in an MEC network with LEO satellites. However, these studies only considered a single service request and largely ignored the network function chaining; thus, their solutions cannot be applied to the service chaining problem in an SEC network.

There are studies on service chaining in both LEO networks and SEC networks [5], [19], [20], [31], [32], [33]. Most of them however either focused on satellite communication networks without the considering of edge computing technology, or ignored the hierarchical structure of an SEC network. Also, none of them considered the uncertainty of service delays in a fully dynamic SEC network. For example, Gao et al. [19] studied the Virtualized Network Function (VNF) placement problem in a satellite network and proposed a solution based on a potential game to maximize the overall network payoff. Another work by Gao et al. [20] investigated the dynamic

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 13,2024 at 13:29:56 UTC from IEEE Xplore. Restrictions apply.

resource allocation problem for VNFs in satellite networks, and proposed a distributed VNF placement algorithm to minimize the network bandwidth cost and the service end-to-end delay jointly. Jia et al. [31] exploited a time-evolving graph to capture the intermittent but predictable connections between mutually visible satellites of a satellite network. They investigated the problem of cost minimization for VNF placement and routing, subject to network resource capacity constraints [33]. Cai et al. [5] investigated the problem of service function chaining in a satellite network, through proposing an exact solution via an Integer Linear Program (ILP) and devising an efficient heuristic as well. Jia et al. [32] studied the service chaining problem in a software-defined satellite network. Li et al. [36] investigated the online dynamic VNF mapping and scheduling in a space-air-ground-integrated network for the network services for Internet of Vehicles users.

All the above studies focused on single-layer LEO satellite networks. Given the fact that both LEO satellites with different orbit heights and high-altitude satellites exist in the space, there are studies focusing on the placement of VNFs in a multi-layer satellite network, by considering computing units in the space and air, such as UAVs, LEOs, and high-altitude satellites. However, these studies may not be applied directly to the problem of this paper, due to the reasons that (1) placing VNFs into multiple orbits or different layers may cause prohibitive waiting delays, (2) the uncertain service delays play a vital role in guaranteeing the quality of service, and (3) VNFs generate states that need to be synchronized to other VNFs of a service chain. For example, Zhang et al. [60] transformed the resource allocation problem in space-air-ground integrated networks into a multi-domain virtual network embedding problem and proposed a reinforcement learning algorithm for the problem. Another work by Zhang et al. [61] studied the problem of service function chain mapping in space-air-ground integrated networks and designed a prediction method to obtain network delays and devised an efficient embedding algorithm for the problem. Li et al. [37] investigated the problem of service function path selection in multi-layer satellite networks, through proposing a path selection algorithm. Wang et al. [49] formulated the service function chaining planning problem as an integer nonlinear programming and devised a heuristic greedy algorithm for it. Cai et al. formulated the problem of service function chains deployment in multi-layer satellite networks as an an integer linear programming and proposed a heuristic algorithm to solve it. Maity et al. [39] studied the problem of virtual network embedding in non-terrestrial networks and devised a dynamic algorithm to maximize the service acceptance rate and revenue. Qin et al. [42] formulated the problem of multiple service function chains embedding in the ultra-dense LEO satellite-terrestrial integrated network as a non-cooperative game and designed a response algorithm with faster convergence and an adaptive play algorithm with more capacity for solutions.

III. Preliminaries

In this section, we first introduce the SEC network model, satellite communication and service models. We then give the definition of the optimization problem precisely.

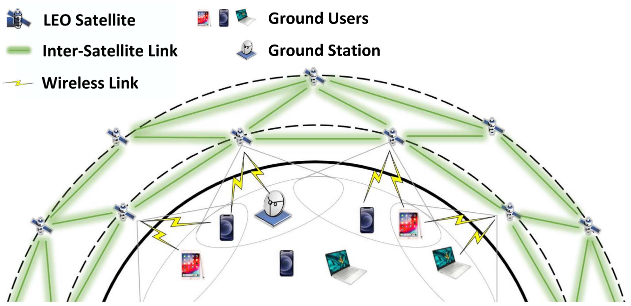


Fig. 1. Example of an SEC network.

A. System Model

We consider a hierarchical SEC network $G = (V \cup U, E)$ with satellites distributed into orbits with different heights, where V is a set of LEO satellites, U is a set of ground users, and E is a set of links among the LEO satellites and between the ground users and the LEO satellites. Each LEO satellite circulates the Earth in an orbit at a given height to provide various network services to its ground users. Let $v_j \in V$ be an LEO satellite, with $1 \leq j \leq |V|$. Each LEO satellite v_j is attached with an amount of computing resource (e.g., an edge server, FPGA, or a neural network accelerator) to implement various VNFs as software running in VMs or containers. Due to the size limitation, an LEO satellite has a limited amount of computing resource to implement NFV-enabled requests issued by ground users. Let C_j be the amount of computing resource provided by LEO satellite v_j. The available amount of resource of each LEO satellite is dynamically changing. For clarity, we consider a finite time horizon $\mathcal{T}$ that is equally divided into T equal time slots, and the length of each time slot is τ. We then use $C_{j,t}$ to denote the amount of available computing resource of v_j at the beginning of time slot t.

NFV-enabled requests of ground users are sent to LEO satellites for implementation in the SEC network. Each ground user $u \in U$ can communicate with an LEO satellite v_j via a wireless link established by the receiver on the ground user when satellite v_j is within the user's transmission range. Denoted by $e_{u,j}$ the wireless link between the ground user u and LEO satellite v_j. Moreover, LEO satellites communicate with each other via laser inter-satellite links [8], [23], [30]. Let e be the inter-satellite link in E between two satellites. The connections among LEO satellites and ground users are unchanged and the SEC network topology is quasi-static within each time slot. An illustration of the SEC network is shown in Fig. 1.

B. Coverage Model

Each LEO satellite moves around the Earth following its predefined orbit with a fixed height at a constant angular velocity, periodically. The orbit and coverage model of each satellite in the SEC network are described in the following.

Orbit model: An Earth-centered orbit is defined as a regular, repeating path around along which an LEO satellite in space takes around the Earth. A *low earth orbit* is referred to as an Earth-centered orbit near the planet, often specified

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 13,2024 at 13:29:56 UTC from IEEE Xplore. Restrictions apply.

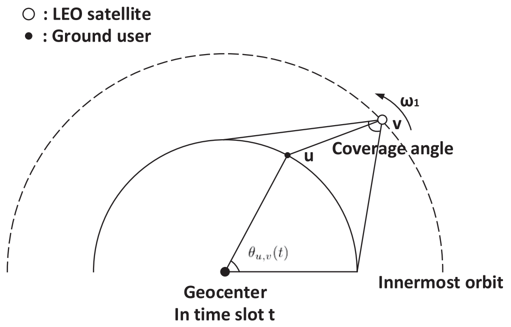


Fig. 2. Illustrative example of the geocentric angle formed by the LEO satellite v coverage boundary and the ground user u at the beginning of time slot t.

as having a period of 128 minutes or less [1]. Denote by $\mathcal{O} = \{o_1, \ldots, o_i, \ldots, o_{|\mathcal{O}|}\}$ a set of low Earth orbits where o_1 is the innermost orbit, o_i is the ith orbit, and $1 \leq i \leq |\mathcal{O}|$. There usually are multiple satellites circulating around each orbit [12], [16], as shown in Fig. 1. Further, the LEO satellites in each orbit are usually evenly distributed around the orbit, and adjacent satellites in each orbit have the same angular distance. Let V_i be the set of LEO satellites which move around orbit o_i. The LEO satellites in V_i move with the same angular velocity ω_i at height h_i and the same angular distance $\frac{2\pi}{|V_i|}$.

Coverage time model: For the sake of communication efficiency, we assume that only the satellites in the innermost orbit o_1 can communicate with ground users directly, as they are the closest satellites to the ground users. Such satellites are referred to as *access satellites*. Each access satellite can only communicate with ground users that are within its coverage area, which is dynamically changing as it moves around its orbit. We assume that each access satellite can cover a ground user for a period with multiple time slots. Without loss of generality, we further consider that there are sufficient number of access satellites such that each ground user can be covered by at least one access satellite at each time slot. Let $\mathcal{V}_t(u)$ be the set of access satellites that the ground user $u \in U$ can communicate in time slot t. The number of time slots of accessing satellite $v_j \in \mathcal{V}_t(u)$ that covers ground user u from the beginning of time slot t can be expressed by

$$T_{u,v}(t) = \left\lfloor \frac{\theta_{u,j}(t)}{\omega_1 \cdot \tau} \right\rfloor, \tag{1}$$

where $\theta_{u,j}(t)$ is the geocentric angle formed by the LEO satellite v_j coverage boundary and the ground user u at the beginning of time slot t, as shown in Fig. 2.

C. Inter-Satellite Links Model

Satellites in an SEC network are interconnected via *inter-satellite links* (ISLs). In each orbit, an LEO satellite connects with two adjacent satellites. Considering that the satellites in the same orbit have the same angular velocity, adjacent satellites in each orbit are relatively static and the ISLs among them can remain stable. However, due to the angular velocity difference between the LEO satellites in adjacent orbits, the connections can change over time. An LEO satellite can communicate with an LEO satellite of the adjacent orbit only if they are within the transmission of each other. The number of time slots that satellite v_j of orbit o_i and LEO satellite $v_{j'}$ of orbit o_{i+1} stay connected from the beginning of time slot t can be calculated by

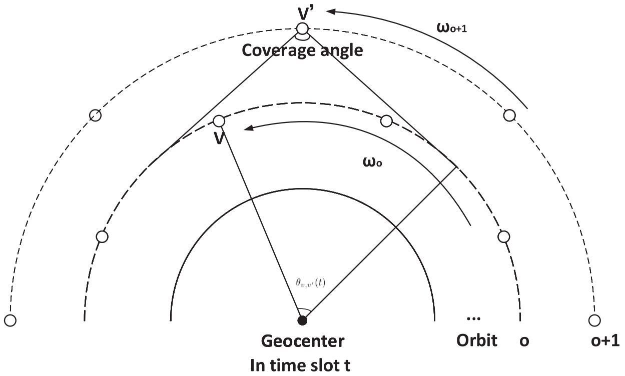


Fig. 3. Illustrative example of the geocentric angle formed by the LEO satellite v_j coverage boundary and satellite $v_{j'}$ at the beginning of time slot t.

$$T_{j,j'}(t) = \left\lfloor \frac{\theta_{j,j'}(t)}{(\omega_i - \omega_{i+1}) \cdot \tau} \right\rfloor, \tag{2}$$

where $\theta_{j,j'}(t)$ is the boundary of the geocentric angle that LEO satellites v_j and $v_{j'}$ can stay connected, as shown in Fig. 3.

D. NFV-Enabled Requests and Service Chains

We consider that ground users are located in remote areas, and they can only communicate by connecting to the SEC network. Applications in such scenario include environment and ship health monitoring at the sea or IoT devices in remote areas. The ground users of such applications transmit their data to LEO satellites in the SEC network for further processing and analysis. To this end, a ground user issues a request that transfers an amount of data traffic from itself to a destination node of the SEC network. To guarantee the security requirement of such applications, the data of each ground user needs to be processed by a sequence of virtualized network functions (VNFs), such as Firewalls and Intrusion Detection Systems (IDS), before reaching its destination. The sequence of VNFs is referred to as a *service chain*, and the request of a ground user is defined as a *NFV-enabled request*.

Let R be a set of NFV-enabled requests issued by the ground users in a finite time horizon $\mathcal{T}$. Each NFV-enabled request $r_k \in R$ is denoted by a quadruple $r_k = (u; d_k; \tau_k; b_k; SC_k)$, where $u \in U$ is the ground user that issues the request r_k as the source node, $d_k \in U$ is the destination node, τ_k is the arrival time slot of request r_k, b_k is the volume of its data traffic, and $SC_k = \langle f_{k,1}, \ldots, f_{k,m}, \ldots, f_{k,L_k} \rangle$ is the service chain of r_k, where $f_{k,m}$ is the mth network function in service chain SC_k and L_k is the number of VNFs in SC_k. We here consider stateful network functions that generate various states during the processing of data traffic. Specifically, the states generated by each VNF $f_{k,m}$ need to be transferred to its next VNF $f_{k,m+1}$ to guarantee that the processing of $f_{k,m+1}$ is correct. For instance, the network functions of Deep Packet Inspection (DPI) [15]

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 13,2024 at 13:29:56 UTC from IEEE Xplore. Restrictions apply.

needs to track information like flow status, protocol status, and application layer status, and pass such state data to the next VNF such as IDS. Following existing studies [44], [59], we assume that the size of states generated by each VNF is usually small and proportional to the data traffic of each request. Let α_k be a constant that is in the range of $[0, 1]$, the volume of the states generated by each VNF of each request r_k thus is $\alpha_k \cdot b_k$.

An implementation of a VNF $f_{k,m}$ in a VM or a container with a certain amount of allocated computing resource of an LEO satellite is considered as an *instance*. Following many existing studies [18], [22], [56], we assume that the computing resource demand of r_k is proportional to its data size that required to be processed by VNFs. That is, the amount of computing resource that is allocated to network function $f_{k,m}$ is

$$C_{unit}(f_{k,m}) \cdot (1+\alpha_k)b_k,$$

where $C_{unit}(f_{k,m})$ is the computing resource to process a unit data by network function $f_{k,m}$.

E. Delay of Requests

The applications of ground users expect a high quality of service, such that their requests are responded timely, by imposing a delay requirement to specify the maximum delay that it can tolerate for transmitting its data from its specified source (ground user) u to the destination node d_k. The delay experienced by a user includes the response delay, transmission delay and the processing delay, which are defined in the following.

Response delay: On the arrival of each NFV-enabled request r_k at time slot τ_k, it needs to be responded as soon as possible. It may be responded immediately by scheduling its data transmission on its arrival, which means that the response delay is 0. Otherwise, the request may be postponed for scheduling due to the availabilities of LEO satellites. This case incurs a non-zero response delay. Let t be the time when NFV-enabled request r_k is scheduled for data transmission and processing, then,

$$t = \sum_{t' \geq \tau_k} x_{k,t'} t', \tag{3}$$

where $x_{k,t'}$ is a binary decision variable the shows whether request r_k is scheduled for implementation in time slot t' after its arrival in time slot τ_k. The response delay of r_k thus is

$$t - \tau_k. \tag{4}$$

Transmission delay: The transmission delay experienced by NFV-enabled request r_k depends on not only the amount of data traffic that needs to be transmitted but also the transmission rate of the links. Let $\delta_t^{unit}(e)$ be the delay of transmitting a unit data via link $e \in E$ in time slot t. Assuming that each NFV-enabled request is scheduled, the transmission delay is the total delay of transmitting data of NFV-enabled request r_k along a selected path p_k in the SEC network from its source s_k to destination d_k, i.e.,

$$\delta_k^{tran} = \sum_{p \in \mathcal{P}^t} z_{k,t}^p \sum_{e \in p} \cdot \delta_t^{unit}(e) \cdot (1+\alpha_k)b_k, \tag{5}$$

where $z_{k,t}^p$ is binary decision variable that indicates the data of NFV-enabled request r_k is routed via path p from the set $\mathcal{P}^t$ of possible paths from s_k to d_k in time slot t. It must be mentioned that a candidate path p meets the following two conditions: (1) the first satellite in p is an access satellite that makes sure the user is connected via satellites in orbit o_1, and (2) the length of the interconnection time among the two satellites of each edge in path p is enough to transmit the data of r_k.

Processing delay: The processing delay experienced by NFV-enabled request r_k depends on the amount of data traffic that needs to be processed and the processing rate of an LEO satellite. Let $\delta_t^{unit}(v_j)$ be the delay of processing a unit data on LEO satellite $v_j \in V$ in time slot t, which depends on the amount of computing resource allocated to process each unit data and the congestion levels of the satellite. The accumulative processing delay incurred due to the data processing by the network functions in service chain SC_k of NFV-enabled request r_k is

$$\delta_k^{proc} = \sum_{f_{k,m} \in SC_k} \sum_{v_j \in V} y_{k,m}^j(t) \cdot \delta_t^{unit}(v_j) \cdot (1+\alpha_k)b_k, \tag{6}$$

where $y_{k,m}^j(t)$ is binary decision variable indicating that whether VNF $f_{k,m}$ of service chain SC_k is deployed in satellite $v_j \in V$ in time slot t.

The delay experienced by NFV-enabled request r_k thus is

$$\delta_k = t - \tau_k + \delta_k^{tran} + \delta_k^{proc}. \tag{7}$$

F. Cost Models

The service provider operating an SEC network wants to minimize its operational cost such that its profit is maximized. The operational cost usually is due to the usage of computing and bandwidth resources, and instantiation costs of VNF instances in LEO satellites. Let $c(v_j)$ and $c(e)$ be the usage costs of one unit of computing and bandwidth resources at LEO satellite $v_j \in V$ and link $e \in E$, respectively. Denote by $c_j(f_{k,m})$ the cost of instantiating an instance of VNF $f_{k,m}$ in LEO satellite $v_j \in V$. The operational cost of the admission of NFV-enabled request r_k thus is

$$c_k = \sum_{f_{k,m} \in SC_k} \sum_{v_j \in V} y_{k,m}^j (c(v_j) \cdot (1+\alpha_k)b_k + c_j(f_{k,m})) + \sum_{p \in \mathcal{P}^t} z_{k,t}^p \sum_{e \in p} c(e) \cdot (1+\alpha_k)b_k.$$

It must be mentioned that although we do not consider the energy consumption of satellites in the SEC network, the proposed model can be easily extended to consider the energy cost that depends on the number of computing cycles used to process data. However, considering the energy consumption of satellites may involve many challenging issues [11], [19], [32], [62]. For example, the energy of a satellite usually is powered by solar panels. How to guarantee the perpetual operational of satellites by considering the charging periods of satellites is challenging. Besides, how to consider an additional dimension of availabilities when the satellites are running out of its energy while still waiting for its next period of charging. We thus consider the energy consumption of satellites as a future topic of this paper.

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 13,2024 at 13:29:56 UTC from IEEE Xplore. Restrictions apply.

G. *Problem Definitions*

Given an SEC network $G = (V \cup U, E)$ and a set of requests R issued by ground users in each time slot t of a finite time horizon T. The data of each NFV-enabled request $r_k \in R$ must be transmitted to its specified destination satellite and processed by the VNFs of its service chain SC_k that is placed in LEO satellites before reaching its destination. We consider the following optimization problems.

The *delay-aware service chaining problem in an SEC network with a single request* is to schedule each request $r_k \in R$ for implementations on its arrival and route the data of each request r_k from its ground user to destination node and instantiate an instance of each VNF in an LEO satellite to process the data of the NFV-enabled request r_k while transmission, with an aim to minimize the delay of implementing a request, subject to the computing resource capacity constraint on each LEO satellite v in each time slot t and the budget B_k on the cost of implementing each request r_k.

The optimization objective of the delay-aware service chaining problem in an SEC network thus can be formulated as

$$(\mathbf{P1}): \ \min \delta_k, \tag{8}$$

subject to the following constraints,

$$t = \sum_{t' \geq \tau_k} x_{k,t'} t', \tag{9}$$

$$\sum_{t' \in \mathcal{T}: t' \geq \tau_k} x_{k,t'} = 1, \tag{10}$$

$$\sum_{f_{k,m} \in SC_k} y^j_{k,m}(t) \cdot C_{unit}(f_{k,m}) \cdot (1 + \alpha_k) \cdot b_k \leq C_{j,t}, \forall v_j, \tag{11}$$

$$\sum_{p \in \mathcal{P}^t} z^p_{k,t} = 1, \tag{12}$$

$$\sum_{v_j \in \Phi(p)} y^j_{k,m}(t) = z^p_{k,t}, \text{for each } f_{k,m} \text{ and } p \in \mathcal{P}^t \tag{13}$$

$$c_k \leq B_k, \tag{14}$$

$$x_{k,t'}, z^p_{k,t}, y^j_{k,m}(t) \in [0, 1], \tag{15}$$

where Inequality (9) determines when the request r_k will be responded after its arrival, and (10) shows that each request is responded for implementation in a single time slot. This means that after time slot t the data of the request will start being processed. Inequality (11) indicates that the computing resource capacity of an LEO satellite cannot be violated. Inequality (12) means that a single path from source s_k to destination d_k in the SEC network has to be selected for request r_k. Eq (13) shows that each VNF $f_{k,m}$ can only be placed to the satellites of the selected path for request r_k, where $\Phi(p)$ is the set of LEO satellites in path p. Inequality (14) shows that the cost of implementing each request r_k cannot meet its budget requirement, denoted by B_k. Constraint (15) shows that the variables are binary decision variables.

The aforementioned optimization problem **(P1)** is NP-Hard. If we consider a special case of **(P1)** that schedules the user requests immediately without waiting, the VNFs of each request is consolidated into a single satellite, and there is no budget requirement, the problem can be transferred to a general assignment problem (GAP). Since the GAP problem is NP-Hard [10], [29] and **(P1)** is a general version of GAP, the problem **(P1)** is NP-Hard as well.

The delay of transmission and processing usually depends on various factors, such as congestion levels of links and status of satellites. As such, the arrival of NFV-enabled requests and the delay are uncertain. The *online service chaining problem in an SEC network with uncertain delays* is to schedule each currently-arrived request for implementation, and route the data of each admitted request $r_k \in R$ from its ground user to destination node and instantiate an instance of each VNF in LEO satellite to process the data of each request r_k while transmission, with an aim to minimize the average delay of each implemented NFV-enabled request, subject to the computing resource capacity constraint on each LEO satellite v_j in each time slot t and the budget B_k on the cost of implementing each request r_k.

For clarity, the symbols used in this paper are summarized in Table I.

IV. Approximation Algorithm for the Delay-Aware Service Chaining in an SEC Network With a Single Request

We now propose an approximation algorithm for the delay-aware service chaining in an SEC network with a single request.

A. *Overview*

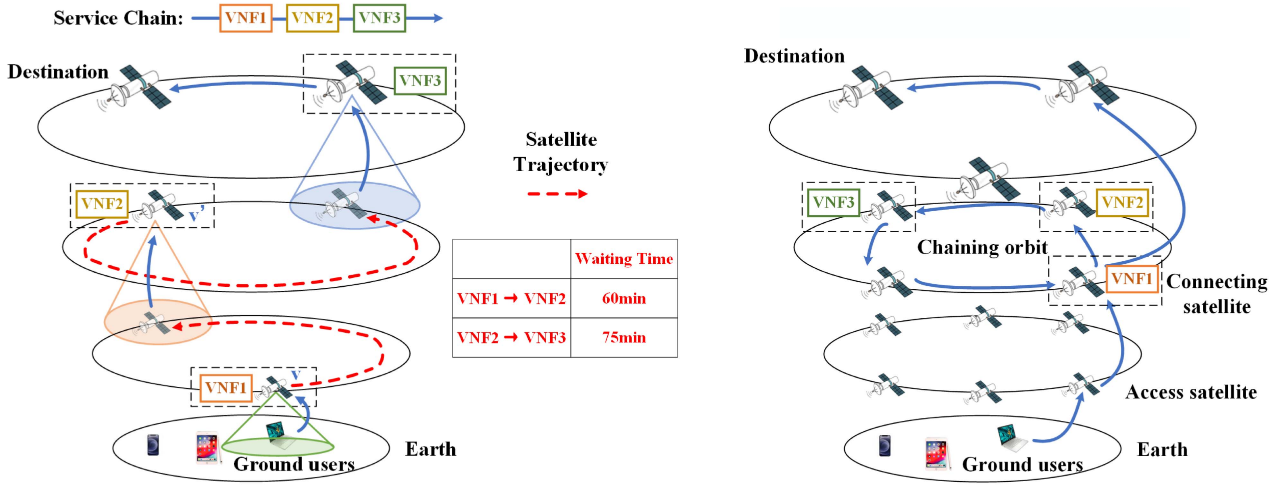
Given a NFV-enabled request r_k arrived at time slot τ_k, it needs to be scheduled to transmit its data from ground user u to a destination d_k, such that its data is processed by the specified VNFs before reaching the destination. To this end, the data of request r_k needs to be transmitted to an access satellite that is currently within its transmission range, and then forwarded to an instance of the service chain before reaching its destination. Intuitively, the VNFs of each NFV-enabled request may be distributed into satellites in any orbit. However, as shown in Fig. 4(a), when LEO satellites v and v' in different orbits are not in each other's transmission range, the data transmission among the two VNFs placed in them has to wait or be forwarded by other satellites, leading to longer communication paths and higher delays.

Motivated by the afore-mentioned problem, we observe that the relative distances of satellites within each orbit are quasi-static, and the network topology of the satellites within an orbit is stable. We thus assume that the VNFs of each service chain SC_k are placed to the LEO satellites in a single orbit, which is referred to as *chaining orbit*, instead of distributing VNFs into different orbits. Fig. 4(b) shows the basic idea of the chaining orbit. Specifically, once the VNFs of a service chain are placed into satellites within a chaining orbit, the data of the request will

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 13,2024 at 13:29:56 UTC from IEEE Xplore. Restrictions apply.

TABLE I
SYMBOLS

Symbols	Meaning
$G = (V \cup U, E)$	a satellite edge computing (SEC) network with a set V of LEO satellites, a set U of ground users and a set E of links
R, $r_k = (s_k; d_k; b_k; SC_k)$	a set of NFV-enabled requests, an NFV-enabled request where $s_k \in U$ is the ground user as the source, $d_k \in U$ is the destination node, b_k is the volume of its data traffic, and SC_k is the service function chain of r_k
$f_{k,m}$	the m-th network function in service chain SC_k
$t, \mathcal{T}, T$	time slot, time horizon, total number of time slot in the finite time horizon $\mathcal{T}$
$C_j, C_{j,t}$	the amount of computing resource of LEO satellite v_j, and the amount of available computing resource at the beginning of time slot t of LEO satellite v_j
$\mathcal{O}, o_i$	a set of LEO satellite orbits and an orbit o_i, where o_1 is the innermost orbit and $o_{\lvert\mathcal{O}\rvert}$ is the outermost orbit
V_i, ω_i	a set of LEO satellites which move around orbit o_i, and the angular velocity of the LEO satellite $v_j \in V_i$
$h_i, \mathcal{V}_t(u)$	the height of the LEO satellite v, and a set of access satellites that the ground user $u \in U$ can access in time slot t
$\theta_{u,j}(t)$	the geocentric angle formed by the LEO satellite v_j coverage boundary and the ground user u at the beginning of time slot t
$\theta_{j,j'}(t)$	the geocentric angle formed by the LEO satellite $v_{j'}$ coverage boundary and the LEO satellite v_j in time slot t
$T_{u,v}(t)$	the number of time slots that access satellite $v_j \in \mathcal{V}_t(u)$ covers ground user u from the beginning of time slot t
$T_{j,j'}(t)$	the number of time slots that LEO satellite v_j of orbit i and LEO satellite $v_{j'}$ of orbit $i+1$ stay connected from the beginning of time slot t
$C_{unit}(f_{k,m})$	the computing resource to process a unit data by $f_{k,m}$
$\delta_t^{unit}(e)$, δ_k^{tran}	the delay of transmitting a unit data via link $e \in E$ in time slot t, and the transmission delay of request r_k
$\delta_t^{unit}(v_j)$, δ_k^{proc}	the delay of processing a unit data on LEO satellite $v_j \in V$ in time slot t, and the processing delay of request r_k
$\mathcal{P}^t$, p, $\Phi(p)$	a set of possible paths from s_k to d_k in time slot t, a path p from $\mathcal{P}^t$, and a set of LEO satellites in path p
$x_{k,t'}$	the binary decision variable, which indicates request r_k is scheduled for implementation in time slot t' after its arrival in time slot τ_k
$z^p_{k,t'}$	the binary decision variable, which indicates the data of NFV-enabled request r_k is routed via path p in time slot t
$y^j_{k,m}(t)$,	the binary decision variable, which indicates VNF $f_{k,m}$ is deployed in LEO satellite $v_j \in V$ in time slot t
δ_k	the delay experienced by request r_k
$c(e), c(v_j)$	the usage cost of one unit of bandwidth and computing resource at link $e \in E$ and LEO satellite $v_j \in V$
$c_j(f_{k,m})$	the cost of instantiating an instance of VNF $f_{k,m}$ in LEO satellite $v_j \in V$
B_k	the budget of implementing NFV-enabled request r_k



	Waiting Time
VNF1 → VNF2	60min
VNF2 → VNF3	75min

(a) Service chaining in multiple orbits.

(b) Service chaining in a chaining orbit.

Fig. 4. Motivation of chaining orbits. (a) shows the case that the transmission delays among VNFs of a service chain may be prohibitively long if the VNFs are placed in LEO satellites in different orbits. The reason is that the LEO satellites in different orbits may wait long time when they move into the transmission range of each other; (b) is an illustrative example of a chaining orbit that placed the VNFs of a service chain. A NFV-enabled request transmits its data from a ground user to access satellite, and then its data is forwarded to chaining orbit and processed by the specified service chain, finally forwarded to its destination.

be forwarded from its selected access satellite to the connecting satellite of the chaining orbit. Afterwards, the data will be processed by the VNFs within the chaining orbit following the predefined order of its service chain. After processing by all the VNFs the processed data is forwarded from the satellite with the last VNF in the service chain to the destination via finding a shortest path in the SEC network. The benefit of placing the VNFs of a service chain in satellites of a chaining orbit is to reduce the prohibitive data transmission delays between two VNFs that are located in different orbits.

Solving the problem then is to first find an access satellite and select a chaining orbit for request r_k, and then place the VNFs of the service chain SC_k of request r_k to the satellites of the selected chaining orbit. To this end, the proposed algorithm performs two major stages: (1) **stage 1**: finding an access satellite for NFV-enabled request r_k and selecting an orbit as the chaining

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 13,2024 at 13:29:56 UTC from IEEE Xplore. Restrictions apply.

orbit for r_k; and (2) **stage 2**: chaining VNFs of NFV-enabled request r_k in the chaining orbit.

B. *Stage 1: Jointly Selecting the Access Satellite and Chaining Orbit*

We observe that the access satellite and the chaining orbit need to be jointly selected; otherwise, longer paths may be used to route the data, thereby leading to higher delays. Specifically, the data of NFV-enabled request r_k needs to be forwarded from the access satellite to the chaining orbit and then to the destination node. If the access satellite is far from any satellite in the chaining orbit, the transmission delay will be high. Note that here has to be an LEO satellite in the chaining orbit receiving the data of NFV-enabled request r_k from the access satellite of r_k. Such a satellite, referred to as *connecting satellite*, should be close to both the access satellite and the destination satellite of request r_k, such that the data of r_k is transmitted in low delay. The problem of jointly selecting the access satellite and chaining orbit then is transferred to finding an access satellite and a connecting satellite of different orbits of the SEC network.

The proposed algorithm performs a delay constrained shortest path in an auxiliary graph to jointly find the access satellite and a connecting satellite along with its orbit as the chaining orbit. Let $G' = (V', E')$ be the auxiliary graph. The vertex set V' has four layers:

- **Layer 1** of the auxiliary graph G' consists of the ground user u of r_k, i.e., $V' \leftarrow V' \cup \{u\}$.
- **Layer 2** has the access satellites that are within the transmission range of ground user u of request r_k in different time slots starting from its arrival time τ_k. Specifically, we can create a *virtual access satellite* for each access satellite in each time slot t of the finite horizon T. However, the request can only be scheduled for implementation after its arrival time slot τ_k. Besides, the access satellite needs to be within its transmission range. Therefore, if a satellite v is within the transmission range of r_k in time slot t with $t \geq \tau_k$, we create a virtual access satellite for it. This means that if the satellite moves out of the transmission range of r_k, it will not be considered as a candidate access satellite for r_k. Let $v'_{j,t}$ be such a virtual access satellite in time slot t, we have $V' \leftarrow V' \cup \{v'_{j,t}\}$ for each satellite v_j that is within the transmission range of ground user u in time slot t.
- **Layer 3** is composed of the satellites in V, with each potentially serving as a *connecting satellite* of an orbit, i.e., $V' \leftarrow V' \cup \{v_j\}$.
- **Layer 4** of the auxiliary graph G' has the destination satellite of r_k, i.e., $V' \leftarrow V' \cup \{d_k\}$.

We then connect the ground user u in **layer 1** with each virtual access satellite $v'_{j,t}$ in **layer 2**, which means that the data of NFV-enabled request r_k can be transmitted to satellite v_j in time slot t. The delay consists of the response delay of scheduling request r_k at time slot t and the data transmission delay from the ground user to the selected satellite. Let $\delta(\langle u, v'_{j,t}\rangle)$ be the delay of edge $\langle u, v'_{j,t}\rangle$ in auxiliary graph G', then,

$$\delta(\langle u, v'_{j,t}\rangle) = t - \tau_k + \delta_t^{unit}(e_{u,j}) \cdot b_k, \tag{16}$$

where $e_{u,j}$ is the wireless link that interconnects ground user u and access satellite v_j in time slot t. It must be mentioned that (16) considers the waiting time of a satellite orbiting to the transmission range of r_k, if $t > \tau_k$, which is $t - \tau_k$. The cost of this edge is set to

$$c(\langle u, v'_{j,t}\rangle) = c(e_{u,j}) \cdot b_k. \tag{17}$$

Note that only VNFs generate states, the above cost function only considers the data traffic of r_k.

Each virtual access satellite in **layer 2** may be connected to each satellite $v_{j'}$ in orbit o_i in **layer 3**. Specifically, there is an edge from $v'_{j,t}$ to $v_{j'}$ in V', i.e., $\langle v'_{j,t}, v_{j'}\rangle$, if the available computing resource of the satellites in o_i is enough to implement the VNFs of SC_k. That is,

$$\sum_{f_{k,m} \in SC_k} C_{unit}(f_{k,m}) \cdot (1 + \alpha_k) b_k \leq \sum_{v_{j''} \in V_{o(j')}} C_{j'',t}, \tag{18}$$

where $o(v_{j'})$ is the orbit of the connecting satellite $v_{j'}$. The delay and costs of $\langle v'_{j,t}, v_{j'}\rangle$ are set to the delay and costs of forwarding the data of r_k from access satellite v_j to connecting satellite $v_{j'}$ via path $p_{j,j'}(t)$, i.e.,

$$\delta(\langle v'_{j,t}, v_{j'}\rangle) = \sum_{e \in p_{j,j'}(t)} \delta_t^{unit}(e) \cdot b_k, \tag{19}$$

and

$$c(\langle v'_{j,t}, v_{j'}\rangle) = \sum_{e \in p_{j,j'}(t)} c(e) \cdot b_k, \tag{20}$$

respectively. Although access satellite v_j may move out of the range of satellite $v_{j'}$, they can still be connected via other satellites via path $p_{j,j'}(t)$ with ISLs.

Each connecting satellite $v_{j'}$ in **layer 3** of G' is then connected with the destination satellite d_k of request r_k in **layer 4** of G'. The delay of edge $\langle v_{j'}, d_k\rangle$ is set to the sum of the transmission delay along the path $p_{j',d_k}(t)$ from satellite $v_{j'}$ to destination satellite d_k in time slot t and the processing delay by the satellites in the chaining orbit. However, the actual processing delays are determined only when the VNFs are placed in the satellites of the chaining orbit in **Stage 2** of the algorithm. We here thus use an *estimated delay* $\delta(\langle v_{j'}, d_k\rangle)$ by VNFs in the satellites in the chaining orbit o_i of $v_{j'}$, then

$$\delta(\langle v_{j'}, d_k\rangle) = \sum_{e \in p_{j',d_k}(t)} \delta_t^{unit}(e) \cdot (1 + \alpha_k) b_k + \hat{\delta}_k^{ch}, \tag{21}$$

where $\hat{\delta}_k^{ch}$ is the sum of an estimated processing and transmission delay if the VNFs of SC_k are placed to the satellites of chaining orbit o_i with $v_{j'}$, and the data transmission among the satellites in orbit o_i. We here adopt an adaptive estimation of the delay $\hat{\delta}_k^{ch}$, which is calculated by

$$\hat{\delta}_k^{ch} = (1 - \xi)\delta_{\min,i} + \xi \cdot \delta_{\max,i}, \tag{22}$$

where ξ is a constant parameter with $0 < \xi \leq 1$, $\delta_{\min,i}$ is the minimum sum of the processing delay by VNFs located in the satellites in orbit o_i and transmission delay among the placed VNFs of the service chain of request r_k. Note that $\delta_{\min,i}$ and $\delta_{\max,i}$ are given as a priori, and the real delay depends on

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 13,2024 at 13:29:56 UTC from IEEE Xplore. Restrictions apply.

the service chaining in the satellites of orbit o_i, which will be described in the next subsection. Similarly, the cost of edge $\langle v_{j'}, d_k \rangle$ is set to

$$\hat{c}_k^{ch} = (1-\phi)c_{\min,i} + \phi \cdot c_{\max,i}, \tag{23}$$

where ϕ is a constant parameter with $0 < \phi \leq 1$, $c_{\min,i}$ and $c_{\max,i}$ are the minimum and maximum costs of placing the VNFs in the satellites of the selected chaining orbit and transmitting data among the placed VNFs.

Given the constructed auxiliary graph G', we then find a constrained shortest path from u to d_k in G' regarding to the delay settings of edges in E', with the cost of the path no more than the budget B_k of r_k. Let p' be the found shortest path. Let $v'_{j,t}$ and $v_{j'}$ be the virtual access satellite and the connecting satellite in the shortest path p'. We then will forward the data of r_k from its ground user u to access satellite v_j, and then to the orbit o_i of $v_{j'}$ for being processed by the service chain in o_i. The processed traffic will then be forwarded from $v_{j'}$ to destination d_k.

C. *Stage 2: Service Chaining With a Selected Access Satellite and Chaining Orbit*

Given the selected chaining orbit o_i for NFV-enabled request r_k, we now place the VNFs of its service chain SC_k into the satellites of the chaining orbit o_i. Recall that there is a connecting satellite that is used to receive data from the access satellite of r_k and forward the processed data to its destination node. Let v_j be a connecting satellite of orbit o_i.

The VNFs of SC_k may be consolidated into the connecting satellite v_j, thereby saving the transmission delay among the VNFs. The delay thus is the minimum potential delay $\delta_{\min,i}$ of placing VNFs in orbit o_i, which only consists of processing the data of r_k in satellite v_j, which in the worst case is

$$\delta_{\min,i} = L_k \max_{v_j \in V_i} \delta_t^{unit}(v_j) \cdot (1+\alpha_k) b_k. \tag{24}$$

Similarly, the transmission cost is saved as well, and we have

$$c_{\min,i} = L_k \max_{v_j \in V_i} c(v_j) \cdot (1+\alpha_k) b_k. \tag{25}$$

However, $v_{i,j}^c$ may not have enough computing resource to implement all the VNFs of SC_k. As such, the VNFs of SC_k need to be distributed to the satellites of orbit o_i. Considering that the satellites of o_i are evenly distributed in the orbit o_i and only two adjacent satellites can directly connect to each other, we greedily find the satellites that can place the VNFs of SC_k around the connecting satellite $v_{j,i}^c$. That is, we place the VNFs of request r_k one by one according to the order in its service chain SC_k. For each currently considered VNF $f_{k,m}$, it is placed to the satellites of orbit o_i following a *First-Fit* manner. Specifically, starting from the connecting satellite v_j of orbit o_i, $f_{k,m}$ is placed to the first satellite of orbit o_i that has enough computing resource to implement $f_{k,m}$. The maximum delay of such a case is due to the fact that the last VNF f_{k,L_k} is placed to the satellite that is furthest from the connecting satellite in either direction. Therefore, the data needs to be routed along the circle of the orbit o_i, leading to a maximum delay of

$$\delta_{\max,i} = |V_i| \cdot \delta_{\max}^{unit} \cdot (1+\alpha_k) b_k, \tag{26}$$

where $\delta_{\max}^{unit}(o_i) = \max\{\delta_t^{unit}(e) \mid \forall\, e \in\ E_i\}$ with E_i being the set of inter-satellite links in orbit o_i. Likewise, the maximum cost can be calculated by

$$c_{\max,i} = (1+\alpha_k)(|V_i| \cdot c_{\max,e} \cdot b_k + L_k \max_{v_j \in V_i} c(v_j) b_k), \tag{27}$$

where $c_{\max,e}$ is the maximum cost of transmitting a unit amount of data along an inter-satellite link of the chaining orbit. Fig. 5 shows the rationale behind.

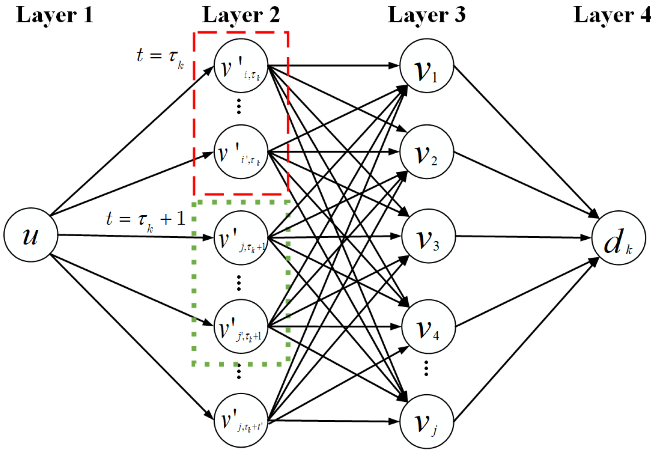


Fig. 5. Example of the auxiliary graph $G' = (V', E')$. **Layer 2** has the *virtual access satellites*, in which the satellites in the red dashed box are within the transmission range of ground user u of request r_k in time slot τ_k and the satellites in the green dotted box are within the transmission range of ground user u of request r_k in time slot $\tau_k + 1$.

Algorithm 1: `ApproSingle`.

```
Input: $G = (V \cup U, E)$, a time horizon $T$, a NFV-enabled
    request $r_k = (u, d_k; \tau_k, b_k, SC_k)$, the capacity
    constraint $C_j$ on each LEO satellite $v_j \in V$ and the
    budget $B_k$ on the cost of implementing request $r_k$.
Output: A minimum delay path of NFV-enabled request $r_k$
    with the capacity and budget constraints, and the
    locations for the VNFs of service chain $SC_k$ of $r_k$.
Construct an auxiliary graph $G' = (V', E')$ as
    illustrated in Fig. 5;
Find a constrained shortest path $p'$ from $u$ to $d_k$ in the
    auxiliary graph $G'$ following Hassin's algorithm [26];
For connecting satellite $v_j$ in the shortest path $p'$, VNFs
    of service chain $SC_k$ are placed one by one on the
    LEO satellites that are close to the connecting satellite
    in the chaining orbit, by using First-Fit algorithm;
Replace each of all edges in $G'$ with its corresponding
    shortest path in network $G$ by Dijkstra algorithm.
```

The detailed procedure of the proposed algorithm is given in **Algorithm** 1, which is referred to as `ApproSingle`.

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 13,2024 at 13:29:56 UTC from IEEE Xplore. Restrictions apply.

D. An Extension

Algorithm ApproSingle jointly schedules each request to a time slot t and places its VNFs to the satellites of a chaining orbit. In particular, it creates a virtual access satellite for each satellite in orbit o_1. However, the selected access satellite may move out of its range in the next time slot. This may lead to the interrupted transmission from a user to the access satellite. To avoid such cases, we now propose an extension to **Algorithm** ApproSingle. Specifically, we slightly modify the set of access satellites that can be included in **layer 2** of the constructed auxiliary graph G'. Recall that in **Algorithm** ApproSingle, if a satellite v is within the transmission range of r_k in each time slot t with $t \geq \tau_k$, we create a virtual access satellite. We now pose an additional constraint on the access satellites whose virtual access satellites can be included into **layer 2**. Specifically, the transmission of data traffic of r_k takes β_k time slots from the request to the satellite, which can be calculated easily given the bandwidth between the request and the satellite. Then, for each time slot t with $t \geq \tau_k$, if a satellite v is within the transmission range of r_k in the period from t to $t + \beta_k$, we create a virtual access satellite. The rest is the same as **Algorithm** ApproSingle. For clarity, we refer to this extended algorithm as **Algorithm** Heu.

E. Algorithm Analysis

We now analyze the correctness and the approximation ratio of **Algorithm** ApproSingle in the following lemmas and theorem.

Lemma 1: Algorithm ApproSingle delivers a feasible solution to the delay-aware service chaining problem in an SEC network, with the computing resource capacity of each satellite being violated by a ratio of $O(|V_{\max}|)$, where $V_{\max}$ is the orbit with the maximum number of satellites.

Proof: Recall that in algorithm ApproSingle, we find a chaining orbit for request r_k, and the VNFs of SC_k will be placed to the satellites of the chaining orbit, thereby consuming computing resource of the satellites. Inequality (18) is used to justify whether the satellites in the chaining orbit have enough computing resource to implement the VNFs of request r_k. However, a satellite in the chaining orbit may not have enough computing resource to implement all VNFs in SC_k, and the VNFs can be distributed into different satellites. Although we can guarantee the total available resource of the satellites in the chaining orbit can meet the total resource demand of SC_k, a single satellite may not be able to fullfil the computing resource demand of a VNF in SC_k. Given the fact that $\sum_{f_{k,m} \in SC_k} C_{unit}(f_{k,m}) \cdot (1+\alpha_k) b_k \leq \sum_{v_{j''} \in V_{o(j')}} C_{j'',t}$ (Inequality (18)), we have that the average amount of available resource of the computing resource of a satellite in the chaining orbit $o(j')$ is at least $(\sum_{v_{j''} \in V_{o(j')}} C_{j'',t})/|V_{o(j')}|$. Therefore, there is at least one satellite in the chaining orbit whose available computing resource is higher than $(\sum_{v_{j''} \in V_{o(j')}} C_{j'',t})/|V_{o(j')}|$, which happens to be the highest amount of available computing resource. In the worst case, all the VNFs are consolidated into such a satellite, leading to a maximum violation ratio of

$$\begin{aligned} &\frac{\sum_{f_{k,m} \in SC_k} C_{unit}(f_{k,m}) \cdot (1+\alpha_k) b_k}{(\sum_{v_{j''} \in V_{o(j')}} C_{j'',t})/|V_{o(j')}|} \\ &\leq \frac{\sum_{v_{j''} \in V_{o(j')}} C_{j'',t}}{(\sum_{v_{j''} \in V_{o(j')}} C_{j'',t})/|V_{o(j')}|} = V_{o(j')} \leq |V_{\max}|. \qquad (28) \end{aligned}$$

We then show the budget violation ratio of algorithm ApproSingle. Recall that in **stage 1** of the algorithm, we find a constrained shortest path in the constructed auxiliary graph G', by setting the cost of edge $\langle v_{j'}, d_k \rangle$ to the estimated cost of service chaining in the orbit of the selected connecting satellite $v_{j'}$. However, in **stage 2** of the algorithm, the real implementing cost depends on where the VNFs are placed in the satellites of the selected chaining orbit. That is, we may select the chaining orbit according to an estimated cost of $c_{\min,i}$ with $\phi = 0$; while the cost of the selected orbit has the maximum cost. As such, the cost budget can be violated by a factor of

$$\begin{aligned} &\frac{c_{\max,i}}{(1-\phi)c_{\min,i} + \phi c_{\max,i}} = \frac{1}{(1-\phi)\frac{c_{\min,i}}{c_{\max,i}} + \phi} \\ &= \frac{1}{(1-\phi)\frac{L_k \max_{v_j \in V_i} c(v_j) \cdot (1+\alpha_k) b_k}{|V_i| c_{\max,e} b_k + L_k \max_{v_j \in V_i} c(v_j) \cdot b_k} + \phi} \\ &= \frac{1}{(1-\phi)\frac{1}{|V_i| c_{\max,e}(1+\alpha_k) b_k / (L_k \max_{v_j \in V_i} c(v_j) \cdot (1+\alpha_k) b_k) + 1} + \phi} \\ &\leq \frac{1}{(1-\phi)\frac{1}{c_{\max,e}/(\max_{v_j \in V_i} c(v_j)) + 1} + \phi}, \text{since } |V_i| \geq L_k \qquad (29) \\ &= \frac{1}{(1-\phi)\frac{1}{\rho+1} + \phi}, \text{ since } |V_i| \geq L_k, \qquad (30) \end{aligned}$$

where ρ is the ratio of the maximum cost of processing a unit amount of data in an edge of the SEC network to the maximum cost of processing a unit amount of data in a satellite in the network, i.e., $\rho = c_{\max,e}/(\max_{v_j \in V_i} c(v_j))$. Therefore, the budget on the cost of implementing request r_k may be violated by a ratio of $\frac{1}{(1-\phi)\frac{1}{\rho+1} + \phi}$. □

Theorem 1: The approximation ratio of the proposed approximation algorithm ApproSingle for the delay-aware service chaining in an SEC network with a single request is $O(\frac{(1+\epsilon)(\kappa+1)^2}{(1-\xi)+(\kappa+1)\xi})$, where κ is the ratio of the maximum delay of processing a unit amount of data in an edge node of the SEC network and the maximum delay of processing a unit amount of data in a satellite in the network. The running time of the proposed algorithm is $O(T\mathcal{V}_{\max}(u) + |V|T\mathcal{V}_{\max}(u) + |V|)(2 + T\mathcal{V}_{\max}(u) + |V|)(\log\log 3 + 1/\epsilon))$.

Proof: We now show the approximation ratio of the proposed algorithm ApproSingle. The objective of the delay-aware service chaining in an SEC network is to minimize the delay of implementing request r_k, which includes the response delay, transmission delay and processing delay. In the proposed algorithm ApproSingle, we find a restricted shortest path in the constructed auxiliary graph. Due to the algorithm by

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 13,2024 at 13:29:56 UTC from IEEE Xplore. Restrictions apply.

Hassin [26], the obtained solution to the restricted shortest path is an $\epsilon-$approximate solution. Specifically, let δ' be the delay of the restricted shortest path and δ'_{opt} be the optimal delay of the restricted shortest path in the auxiliary graph. We have

$$\delta'/\delta'_{opt} = 1+\epsilon. \tag{31}$$

However, d' may not be the actual cost of implementing request r_k, because we use the estimated delay of implementing request r_k in a chaining orbit. That is, we may select the chaining orbit according to an estimated delay of $\delta_{\min,i}$ with $\xi = 0$; while the delay of the selected orbit has the maximum delay. Let δ'' be the actual delay of implementing r_k. We have

$$\begin{aligned}\frac{\delta''}{\delta'} &= \frac{\delta_{\max,i}}{(1-\xi)\delta_{\min,i}+\xi\delta_{\max,i}} = \frac{1}{(1-\xi)\frac{\delta_{\min,i}}{\delta_{\max,i}}+\xi}\\ &= \frac{1}{(1-\xi)\frac{L_k\max_{v_j\in V_i}\delta_t^{unit}(v_j)\cdot b_k}{|V_i|\delta_{\max}^{unit}(1+\alpha_k)b_k}+\xi}\\ &\le \frac{1}{(1-\xi)\kappa+\phi},\ \text{since } |V_i|\ge L_k,\end{aligned} \tag{32}$$

where κ is the ratio of the maximum delay of processing a unit amount of data and the minimum delay of processing a unit amount of data. We then have

$$\frac{\delta''}{\delta'_{opt}} \le \frac{\delta'\frac{1}{(1-\xi)\kappa+\phi}}{\delta'_{opt}} \le \frac{1+\epsilon}{(1-\xi)\kappa+\phi}. \tag{33}$$

Denote by δ_{opt} the optimal solution to the delay-aware service chaining problem in an SEC network. Note that the optimal solution may not place all VNFs to the satellites of a single satellite. We thus have

$$\delta'_{opt} \ge \delta_{opt}. \tag{34}$$

We observe that the main reason for the higher value of δ'_{opt} is the restriction of placing all VNFs in a single chaining orbit. In the ideal case, the optimal solution δ_{opt} places the VNFs solely to the shortest path p^* from the ground user to the destination satellite. According to the construction of the auxiliary graph, such a path p^* corresponds to a shortest path p' from u to d_k in the auxiliary graph, as any satellite can be a potential connecting satellite. That is, the path p' is from the access satellite to its destination via the connecting satellite. This means that the optimal solution δ_{opt} does not place any VNF to a satellite that is not in the path, leading to a delay of $\delta_{\min,i}$ for consolidating the VNFs in the connecting satellite. We thus have

$$\delta'_{opt} \le \frac{\delta_{\max,i}}{\delta_{\min,i}}\delta_{opt} = (1/\kappa)\delta_{opt}. \tag{35}$$

Then,

$$\frac{\delta''}{\delta_{opt}} \le \frac{\delta''}{\kappa\delta'_{opt}} \le \frac{1+\epsilon}{(1-\xi)\kappa^2+\phi\kappa}. \tag{36}$$

We analyze the running time of the proposed algorithm `ApproSingle` as follows. By using Hassin's algorithm [26], the solution of the constrained shortest path in the auxiliary graph $G' = (V', E')$ takes $O(|E'||V'|(\log\log|\frac{UB}{LB}| + 1/\epsilon))$ time, where $|V'|$ is the sum of four layers node numbers, $|E'|$ is the number of fully connected edges between layers, UB and LB are upper and lower bounds for this problem. Consider that the auxiliary graph has only four layers, a path has at most three edges. Therefore, using the sum of the 3 longest edges as a trivial upper bound and 1 as the lower bound. Then we can get that the running time of finding the restricted shortest path in the auxiliary graph $G' = (V', E')$ is $O(|E'||V'|(\log\log 3 + 1/\epsilon))$. In addition, **layer 1** and **layer 4** of auxiliary graph G' both have one node that represents the ground user and destination, respectively. Let $\mathcal{V}_{\max}(u)$ be the maximum number of access satellites that can be accessed by ground user u in the finite time horizon T. Then, the number of nodes in layer 2 is the sum of the number of access satellites in each time slot since the request arrives, which is no more than $T\mathcal{V}_{\max}(u)$. **Layer 3** is composed of all LEO satellites in V. Therefore, $|V'|$ is no more than $(2 + T\mathcal{V}_{\max}(u) + |V|)$. Correspondingly, $|E'|$ is no more than $(T\mathcal{V}_{\max}(u) + |V|T\mathcal{V}_{\max}(u) + |V|)$. □

V. An Algorithm for the Online Delay-Aware Service Chaining in an SEC Network

We now consider the online delay-aware service caching in an SEC network with uncertain delays and the requests arriving into the system one by one without the knowledge of their future arrivals.

A. *Online Algorithm*

Given the uncertain delays of the network and the unknown arrivals of requests, we now devise an adaptive online algorithm that dynamically admits and implements each arrived request one by one. Our idea is to determine the admission of each NFV-enabled request on its arrival by invoking the proposed algorithm `ApproSingle`. However, algorithm `ApproSingle` implements each request according to the estimated delays and costs of the satellites of each orbit, as shown in (22) and (23). Specifically, parameters ξ and ϕ are used to capture the current status of the satellites of a chaining orbit. When the requests arrive into the system dynamically, it complicates the problem by adding an additional dimension of uncertainty, meaning that the status of the satellites can change very frequently. As such, we dynamically adjust the estimated delays and costs on the implementation of an admitted request. Specifically, if the satellites in an orbit already have high workloads, ξ and ϕ can be set to higher values; otherwise, they are set to lower values. On the other hand, smaller values of ξ and ϕ may lead to higher resource violations, because smaller values mean conservative estimation of the delays and costs. In addition, we observe that each request may also prefer different values of ξ and ϕ, since it may have a different resource demand with other requests.

Given the afore-mentioned motivations, we propose an online heuristic that dynamically adjusts the values of ξ and ϕ for each request on its arrival. Specifically, we set a multiple-level waiting queue for the arrived requests, as shown in Fig. 6. Each level of the queue has a different combination of the values of ξ and ϕ. Let q be the qth level of the queue, with $1 \le q \le Q$. The value range of ξ and ϕ is also divided into Q levels. As such, each level q of the queue is associated with ξ and ϕ that are in the range

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 13,2024 at 13:29:56 UTC from IEEE Xplore. Restrictions apply.

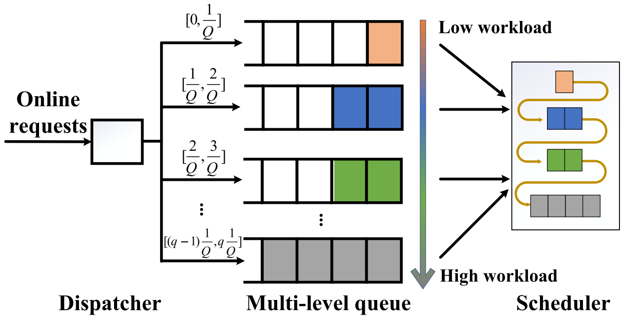


Fig. 6. Example of multi-level queue. The requests are put into different queues according to the current workload status, and then processed by the scheduler in sequence according to the queue order.

of

$$\left[(q-1)\frac{1}{Q}, q\frac{1}{Q}\right]. \tag{37}$$

There is also a dispatcher that dispatches arrived requests to different levels of the queue. In addition, there is a scheduler that schedules the requests of the multiple-level queue, as shown in Fig. 6.

We can see that lower levels of the queue have smaller values of ξ and ϕ, which means that satellites have low workloads. This usually leads to lower delays. However, if the estimation is not accurate, the resource violation can be large as well. On the other hand, higher levels of the queue have larger values of ξ and ϕ, which leads to higher delays since the satellites are usually congested. This setting however is over pessimistic, thereby leading to under utilization of the resources of satellites. Therefore, we consider that the dispatcher assigns requests to the levels of the queue by jointly considering the obtained delay and resource violation ratios. Specifically, we maintain an *active level* of the queue, which means that the currently arrived request will be assigned to the current active level. We increase the active level by one level, if the current resource violations keep increasing for a given number of time slots. Instead, we decrease the active level by one level, if the average delay of implementing each request keeps increasing for the given number of time slots. Meanwhile, the scheduler takes a request from the lowest level first. The requests in higher levels will not be scheduled until all the requests in lower levels are scheduled. Each request is implemented via invoking algorithm `ApproSingle`. The detailed steps of the proposed algorithm are shown in **Algorithm** 2, which is referred to as **Algorithm** `Online`.

Algorithm 2: `Online`.

```
Input: $G = (V \cup U, E)$, a time horizon $\mathcal{T}$, a set of
    NFV-enabled requests that arrive at the SEC network
    with each request $r_k = (u, d_k; \tau_k, b_k, SC_k)$, a given
    number $\hat{t}$ of time slots.
Output: The admission or rejection of each incoming
    NFV-enabled request, if admitted, a minimum delay
    path for the request will be delivered.
for each incoming request $r_k$ do
    if request $r_k$ coming in a new time slot. then
        for $q \leftarrow 1$ to $Q$ do
            Find a minimum delay path $p'$ for each
            request $r_{k'}$ in $q$ level queue by using algorithm
            ApproSingle;
            if $p' \neq \emptyset$ then
                Admit request $r_{k'}$;
            else
                Reject request $r_{k'}$.
            end if
        end for
        Clear all requests in the multi-level queue.
    end if
    Maintain an active level $q$ of queue for each
    request $r_k$;
    if the current resource violations keep increasing
    for $\hat{t}$ then
        $q \leftarrow q + 1$;
    else if the average delay of implementing each
    request keeps increasing for $\hat{t}$ time slots then
        $q \leftarrow q - 1$;
    end if
    According to the active level $q$, push the request $r_k$
    in the corresponding multi-level queue;
end for
```

B. Algorithm Analysis

Theorem 2: Algorithm `Online` for the online delay-aware service chaining in an SEC network delivers a feasible solution in time $O(|R|(T\mathcal{V}_{\max}(u) + |V|T\mathcal{V}_{\max}(u) + |V|)(2 + T\mathcal{V}_{\max}(u) + |V|)(\log\log 3 + 1/\epsilon))$.

Proof: To show the solution delivered by Algorithm `Online` is feasible, we need to show the classification of active level of requests does not affect the solution feasibility of Algorithm `ApproSingle`. Assume that Algorithm `Online` currently considers request r_k. If its previous request r_{k-1} is admitted, the computing resource capacity of the corresponding LEO satellites that implement the VNFs of r_{k-1} is then updated, since the resource availabilities of these LEO satellites have changed. Otherwise, there is no changes of the nodes and edges in the auxiliary graph. Considering that the feasibility of delivering a feasible solution by Algorithm `ApproSingle` can be shown by Lemma 1, Algorithm `Online` delivers a feasible solution when multiple requests are considered.

We then analyze the running time of the proposed algorithm `Online`. Clearly, in the worse case, all the requests arrive in the same time slot. That means we need to find the minimum delay paths for these requests by using $|R|$ times algorithm `ApproSingle`. Therefore, the running time of the proposed algorithm `Online` is $O(|R||E'||V'|(\log\log 3 + 1/\epsilon))$,where $|V'| = 2 + T\mathcal{V}_{\max}(u) + |V|$ and $|E'| = T\mathcal{V}_{\max}(u) + |V|T\mathcal{V}_{\max}(u) + |V|$.

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 13,2024 at 13:29:56 UTC from IEEE Xplore. Restrictions apply.

VI. Simulations

In this section, we evaluate the performance of the proposed algorithms through experimental simulation.

A. Environment Settings

According to the structure of LEO satellite constellations of Iridium II [12], Starlink [16], Globalstar [13] and OneWeb [27], we consider an SEC network consisting of 4 orbits, where the number of LEO satellites in each orbit is in the range of $[6, 10]$ and the altitude of each orbit varies from 780 to 1,414 kilometers. The usage cost of bandwidth resources is 0.01 USD per Gigabyte [64]. There is an edge server in each LEO satellite. The computing resource capacity of each server is randomly selected from 16, 32, 48, and 64 vCPU, and the corresponding costs of using such resources to process unit data are 0.0253, 0.04493, 0.08987, 0.17973 US dollars per second [2]. The locations of ground users in the SEC network are randomly generated and there are four ground users. The source of each NFV-enabled request is randomly selected from the four users, and the destination of the request is randomly selected from the SEC network. The budget of implementing each NFV-enabled request is set to 1.5 USD. The data of each request is randomly drawn from $[1, 50]$ Megabyte, and the delay requirement is randomly generated from $[1.5, 2.5]$ seconds. Besides, the number of VNFs of each NFV-enabled request varies from 5 to 10 [52]. Without otherwise specified, the afore-mentioned settings will be default parameter settings in our simulations. We implemented our simulation in Python and used python package `Networkx` [25] to construct the SEC networks. The result in each figure is based on the average of 50 runs of the proposed algorithms and their benchmarks. The running times of the algorithms are obtained based on a server with 2.80 GHz Intel i7-7700HQ CPU and 16 GB memory.

Benchmark algorithms: We compare the performance of the proposed algorithms against the following benchmarks.

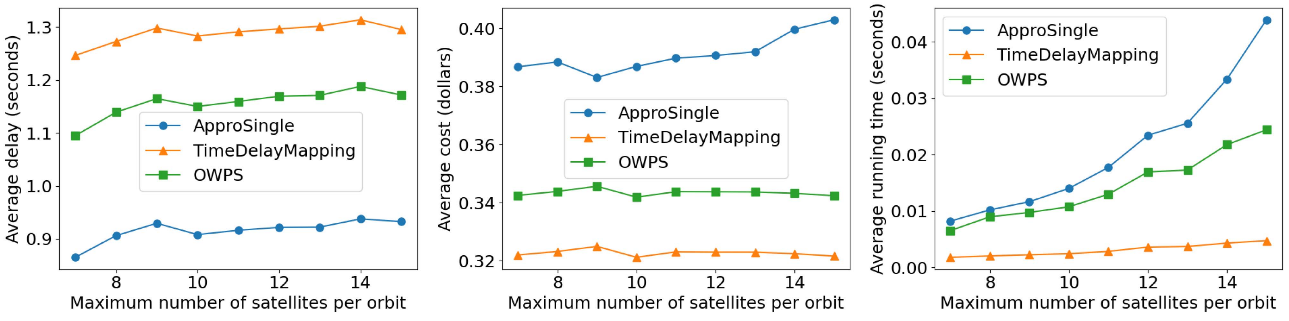
1) `TimeDelayMapping` [61]: Algorithm `TimeDelayMapping` classifies NFV-enabled requests based on their data sizes and delay requirements, and then uses the K-shortest path (KSP) algorithm to select the path with the lowest delay from the paths that meet the delay requirements.
2) `OWPS` [37]: For each NFV-enabled request, algorithm `OWPS` calculates the weights of different parameters of each path (e.g., transmission delay, satellite remaining computing resource, hop count, processing delay, residual connection time) using the entropy weighting method, and uses the theory of grey systems to calculate the grey relationship among the parameters of each path. The optimal path with the optimal weight is then selected.
3) `RL-BA-VNA` [60]: algorithm `RL-BA-VNA` uses reinforcement learning to select the optimal path for each NFV-enabled request. Specifically, `RL-BA-VNA` extracts feature matrix from the hierarchical SEC network and inputs it into the policy network, which outputs the placement probability of VNF on each satellite. Then, it places VNF based on the obtained probability. Here the policy network consists of a revolutionary layer, a RELU exception layer, a fully connected layer, a softmax layer, and a node filtering layer.

Note that the afore-mentioned benchmark algorithms may place VNFs of a service chain into multiple orbits. In contrast, our algorithms adopt the concept of chaining orbit and place the VNFs of each service chain of a single orbit to avoid prohibitively long delays.

B. Performance Evaluation of Algorithm `ApproSingle`

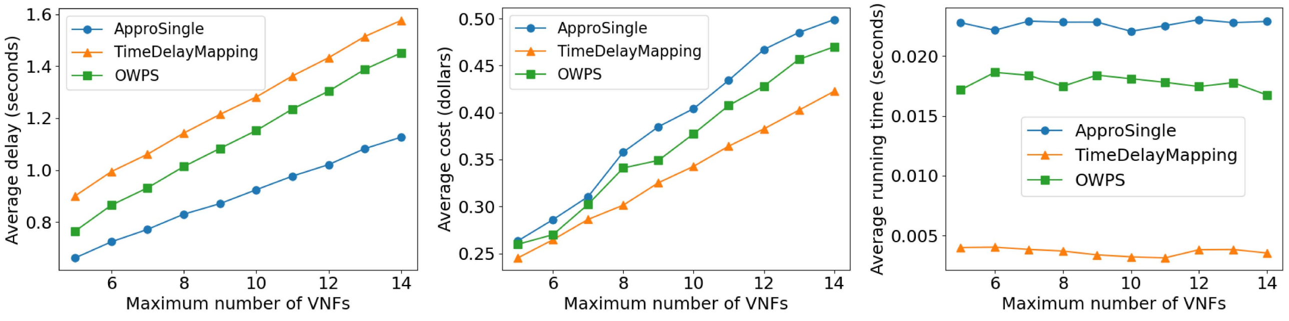
We first evaluate the performance of algorithm `ApproSingle` against that of algorithms `TimeDelayMapping` and `OWPS` in terms of average delay and cost, and average running time, by varying the maximum number of LEO satellites per orbit from 6 to 15 and fixing the number of NFV-enabled requests to 50 and the maximum number of VNFs of each service chain to 10. Note that the running times represent the time taken by the algorithms to make implementation decisions for each NFV-enabled request. Fig. 7 shows the results. It can be seen from Fig. 7(a) that algorithm `ApproSingle` achieves lower delay than that of algorithms `TimeDelayMapping` and `OWPS`. The reason is that algorithm `ApproSingle` places the service chain in the same orbit, reducing the impact of dynamic topology in the satellite network, while algorithms `TimeDelayMapping` and `OWPS` place the service chain in different orbits. Although the transmission delay is lower than that of algorithm `ApproSingle`, due to changes in the topology of the Satellite Network, the waiting delay is increased, resulting in a total delay higher than that of algorithm `ApproSingle`. Besides, algorithm `OWPS` takes into account the remaining connection time of the link when selecting the path, so the delay is relatively low. Furthermore, as shown in Fig. 7(b), the average cost of algorithm `ApproSingle` is higher than that of algorithms `TimeDelayMapping` and `OWPS`, because `ApproSingle` places the service chain in the same orbit, it increases additional transmission costs. In addition, from Fig. 7(c), we can see that the running time of algorithm `ApproSingle` increases with the increase of the number of LEO satellites, which is slightly higher than that of algorithms `TimeDelayMapping` and `OWPS`. The reason is that algorithm `ApproSingle` needs to construct an auxiliary graph and select paths based on the auxiliary graph, and the scale of the auxiliary graph is larger than the original network topology graph. Although the running times of all three algorithms are around tens of milliseconds and our proposed algorithm is lightly higher than the rest algorithms, the average delay is much lower than that of algorithms `TimeDelayMapping` and `OWPS` as shown in Fig. 7(a).

We then evaluate the performance of algorithms `ApproSingle`, `TimeDelayMapping` and `OWPS` in terms of average delay and average cost, average running time, by varying the maximum number of VNFs of each service chain from 5 to 14 and fixing the number of requests to 50 and the maximum number of satellites per orbit to 10. As shown in Fig. 8(a), the average delay of algorithms `ApproSingle`, `TimeDelayMapping` and `OWPS` is increasing due to the increase in the



(a) Average delay (b) Average cost (c) Average running time

Fig. 7. Performance of algorithms ApproSingle, TimeDelayMapping and OWPS with different maximum numbers of satellites per orbit.



(a) Average delay (b) Average cost (c) Average running time

Fig. 8. Performance of algorithms ApproSingle, TimeDelayMapping and OWPS with different maximum numbers of VNFs.

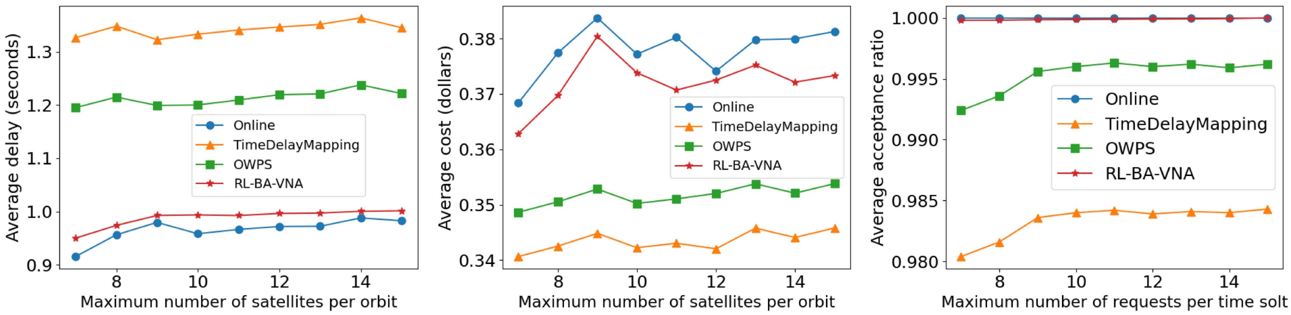
maximum number of VNFs, which is leading to longer processing times. However, it is evident that the average delay growth rate of algorithm ApproSingle is slower compared to that of algorithm TimeDelayMapping and algorithm OWPS. This can be attributed to the fact that as processing delay increases, the impact of changes in satellite network topology will also increase. Nevertheless, algorithm ApproSingle is designed to mitigate this effect by placing the service chain in the same orbit. Besides, it can be seen in Fig. 8(b) that the costs of all algorithms increase with the growth of the maximum number of VNFs, and the cost of algorithm ApproSingle is the highest. The reason is that the VNFs within a service chain with more VNFs can be placed into multiple LEO satellites, thereby increasing the costs of transmitting data among the VNFs. In addition, from Fig. 8(c), we can see that the number of VNFs has little impact on the running times of algorithms ApproSingle, TimeDelayMapping and OWPS. The number of VNFs has no impact on the time complexity of these three algorithms.

C. *Performance Evaluation of Algorithm* Online

We now evaluate the performance of algorithm Online against that of algorithms TimeDelayMapping, OWPS and RL-BA-VNA in terms of average delay and cost, and average acceptance ratio within a finite time horizon of 50 time slots, by varying the maximum number of LEO satellites per orbit from 6 to 15 and fixing the maximum number of VNFs of each service chain to 10. Fig. 9 shows the results in the last time slot of the finite time horizon. Algorithm Online obtains the lowest delay and the highest costs of all the four algorithms. In addition, the delays of algorithms Online and RL-BA-VNA are similar, but the cost of RL-BA-VNA is lower than that of algorithm Online because algorithm Online places the service chain in the same orbit, limiting the opportunity of selecting satellites with lower processing costs in other orbits. It can be seen in Fig. 9(c) that the average acceptance ratios of algorithms Online and RL-BA-VNA are slightly higher than other algorithms, and the average acceptance ratios of algorithms TimeDelayMapping and OWPS gradually increase and eventually stabilize, as the available computing resource in the satellite network increases with the growth of the maximum number of satellites per orbit.

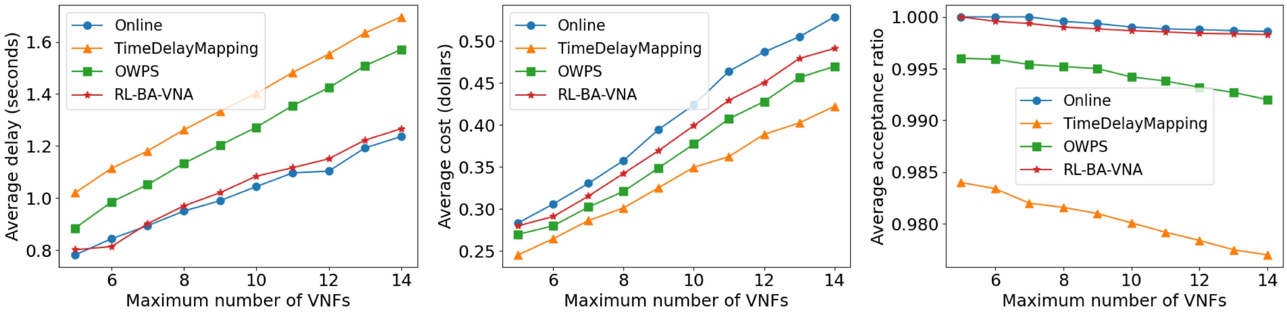
We then evaluate the performance of algorithm Online against that of algorithms TimeDelayMapping, OWPS and RL-BA-VNA in terms of average delay, average cost and average acceptance ratio within a finite time horizon of 50 time slots, by varying the maximum number of VNFs of each service chain from 5 to 14 while fixing the number of requests to 50 and the maximum number of satellites per orbit to 10. The evaluation results are illustrated in Fig. 10. From Fig. 10(a), we can see that the average delay of all algorithms is increasing with the increase of the maximum number of VNFs, leading to longer processing time. It can be seen from Fig. 10(b) that the average cost of all algorithms increases with the growth of the maximum number of VNFs, and the cost of algorithm Online is the highest, the

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 13,2024 at 13:29:56 UTC from IEEE Xplore. Restrictions apply.



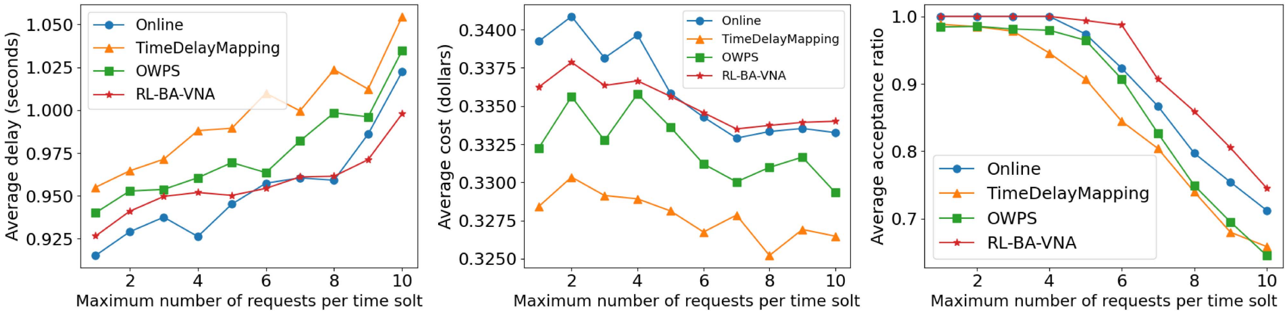
(a) Average delay (b) Average cost (c) Average acceptance ratio

Fig. 9. Performance of algorithms Online, ShortestPath, TimeDelayMapping and RL-BA-VNA with different maximum numbers of satellites per orbit.



(a) Average delay (b) Average cost (c) Average acceptance ratio

Fig. 10. Performance of algorithms Online, ShortestPath, TimeDelayMapping and RL-BA-VNA with different maximum numbers of VNFs in each service chain.



(a) Average delay (b) Average cost (c) Average acceptance ratio

Fig. 11. Performance of algorithms Online, ShortestPath, TimeDelayMapping and RL-BA-VNA with different maximum numbers of requests per time slot.

rationale is that algorithm Online places the entire service chain in a single orbit, which results in increased transmission expenses. Moreover, from Fig. 10(c), we can see that the average acceptance ratios of algorithms Online, TimeDelayMapping, OWPS and RL-BA-VNA decrease with the growth of the maximum number of VNFs. This tendency is due to the diminishing availability of network resources as the maximum number of VNFs grows.

We finally evaluate the performance of algorithm Online against that of algorithms TimeDelayMapping, OWPS and RL-BA-VNA in terms of average delay and cost, and average acceptance ratio by varying the maximum number of NFV-enabled requests per time slot from 1 to 10 and fixing the maximum number of LEO satellites per orbit to 10 and the maximum number of VNFs in each service chain to 10. Fig. 11 shows the results. It can be seen in Fig. 11(a) that algorithm

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 13,2024 at 13:29:56 UTC from IEEE Xplore. Restrictions apply.

`Online` achieves the lowest delay of the four algorithms when the maximum number of NFV-enabled requests per time slot is 8, and algorithm `RL-BA-VNA` performs better when the number of requests per time slot is higher than 8. This is because as the number of requests increases, the number of learning samples gradually increases, and the learning accuracy of algorithm `RL-BA-VNA` also improves. Further, the delay increases as the number of NFV-enabled requests increases. This is because as the number of requests in the same time slot increases, the response delay also increases. Besides, as the the number of NFV-enabled requests keeps increasing, the multi-level queue of algorithm `Online` dynamically adjusts parameters to enable algorithm `Online` to select orbits with more resources for VNF placement. This consequently leads to the increase of processing delay and decreases the cost of algorithm `Online`, as shown in Fig. 11(b). The remaining three algorithms prioritize placing VNFs on satellites with low processing delay, which can result in insufficient resources on those satellites. As a consequence, these algorithms may have to select satellites with high processing delay for VNF placement, leading to increased latency but reduced costs. Furthermore, from Fig. 11(c), we can see that the average acceptance ratios of algorithms `Online`, `TimeDelayMapping`, `OWPS` and `RL-BA-VNA` decrease as the number of NFV-enabled requests grows. This is because as the number of NFV-enabled requests increases, the available resources in the network are decreasing, which can lead to a higher rejection rate.

VII. Conclusion and Future Works

In this article, we investigated the delay-aware service chaining problem in an SEC network. For the problem with a single user request, we proposed an approximation algorithm with an approximation ratio by devising a novel concept of *chaining orbit* and auxiliary graph construction technique. For the online version of the problem with unknown request arrivals and uncertain network delays, we also devised an online algorithm. We conducted extensive simulations to evaluate the performance of the proposed algorithms based on real satellite network topologies. The proposed algorithms outperform its counterparts in terms of both delay and cost.

The future works of this study include (1) we will focus on the space-air-ground integrated MEC networks with highly distributed and heterogeneous computing resource in the network, by developing efficient and effective algorithms for delay-aware service chaining in the network, and (2) we will also investigate the energy consumption of satellites on their resource availability, service chaining performance, through giving a precise energy model for satellites, and devising energy-aware service chaining algorithms.

References

[1] A. Allahvirdi-Zadeh, K. Wang, and A. El-Mowafy, “Precise orbit determination of leo satellites based on undifferenced GNSS observations,” *J. Surveying Eng.*, vol. 148, no. 1, 2022, Art. no. 03121001.

[2] Amazon, “Amazon EC2 on-demand pricing,” 2022. [Online]. Available: https://aws.amazon.com/ec2/pricing/on-demand/

[3] Y. Bi et al., “Software defined space-terrestrial integrated networks: Architecture, challenges, and solutions,” *IEEE Netw.*, vol. 33, no. 1, pp. 22–28, Jan./Feb. 2019.

[4] Y. Borthomieu, “Satellite lithium-ion batteries,” in *Lithium-Ion Batteries*. Elsevier, 2014, pp. 311–344.

[5] Y. Cai, Y. Wang, X. Zhong, W. Li, X. Qiu, and S. Guo, “An approach to deploy service function chains in satellite networks,” in *Proc. IEEE/IFIP Netw. Operations Manage. Symp.*, 2018, pp. 1–7.

[6] X. Cao et al., “Converged reconfigurable intelligent surface and mobile edge computing for space information networks,” *IEEE Netw.*, vol. 35, no. 4, pp. 42–48, Jul./Aug. 2021.

[7] Y. Cao, H. Guo, J. Liu, and N. Kato, “Optimal satellite gateway placement in space-ground integrated networks,” *IEEE Netw.*, vol. 32, no. 5, pp. 32–37, Apr. 2018.

[8] C. Carrizo, M. Knapek, J. Horwath, D. D. Gonzalez, and P. Cornwell, “Optical inter-satellite link terminals for next generation satellite constellations,” in *Free-Space Laser Communications XXXII*. Bellingham, WA, USA: SPIE, 2020, pp. 8–18.

[9] R. Cohen, Liane Lewin-Eytan, Joseph S. Naor, and D. Raz, “Near optimal placement of virtual network functions,” in *Proc. IEEE Conf. Comput. Commun.*, 2015, pp. 1346–1354.

[10] R. Cohen, L. Katzir, and D. Raz, “An efficient approximation for the generalized assignment problem,” *Inf. Process. Lett.*, vol. 100, no. 4, pp. 162–166, 2006.

[11] B. Denby and B. Lucia, “Orbital edge computing: Nanosatellite constellations as a new class of computer system,” in *Proc. 25th Int. Conf. Architectural Support Program. Lang. Operating Syst.*, New York, NY, USA, 2020, pp. 939–954.

[12] B. Di, L. Song, Y. Li, and H. V. Poor, “Ultra-dense LEO: Integration of satellite access networks into 5G and beyond,” *IEEE Wirel. Commun.*, vol. 26, no. 2, pp. 62–69, Apr. 2019.

[13] F. J. Dietrich, P. Metzen, and P. Monte, “The globalstar cellular satellite system,” *IEEE Trans. Antennas Propag.*, vol. 46, no. 6, pp. 935–942, Jun. 1998.

[14] C. Ding, J.-B. Wang, H. Zhang, M. Lin, and G. Y. Li, “Joint optimization of transmission and computation resources for satellite and high altitude platform assisted edge computing,” *IEEE Trans. Wireless Commun.*, vol. 21, no. 2, pp. 1362–1377, Feb. 2021.

[15] R. T. El-Maghraby, N. M. Abd Elazim, and A. M. Bahaa-Eldin, “A survey on deep packet inspection,” in *Proc. 12th Int. Conf. Comput. Eng. Syst.*, 2017, pp. 188–197.

[16] V. L. Foreman, A. Siddiqi, and O. De Weck, “Large satellite constellation orbital debris impacts: Case studies of oneweb and spacex proposals,” in *AIAA SPACE Astronautics Forum Exposition*, Reston, VA, USA: American Institute of Aeronautics and Astronautics, 2017.

[17] J. Fu, J. Hua, J. Wen, K. Zhou, J. Li, and B. Sheng, “Optimization of achievable rate in the multiuser satellite IoT system with SWIPT and MEC,” *IEEE Trans. Ind. Inform.*, vol. 17, no. 3, pp. 2072–2080, Mar. 2021.

[18] T. Gao et al., “Cost-efficient VNF placement and scheduling in public cloud networks,” *IEEE Trans. Commun.*, vol. 68, no. 8, pp. 4946–4959, Aug. 2020.

[19] X. Gao, R. Liu, and A. Kaushik, “Virtual network function placement in satellite edge computing with a potential game approach,” *IEEE Trans. Netw. Service Manage.*, vol. 19, no. 2, pp. 1243–1259, Jun. 2022.

[20] X. Gao, R. Liu, A. Kaushik, and H. Zhang, “Dynamic resource allocation for virtual network function placement in satellite edge clouds,” *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 4, pp. 2252–2265, Jul./Aug. 2022.

[21] S. Geng, S. Liu, Z. Fang, and S. Gao, “An optimal delay routing algorithm considering delay variation in the LEO satellite communication network,” *Comput. Netw.*, vol. 173, 2020, Art. no. 107166.

[22] M. Golkarifard, C. F. Chiasserini, F. Malandrino, and A. Movaghar, “Dynamic VNF placement, resource allocation and traffic routing in 5G,” *Comput. Netw.*, vol. 188, 2021, Art. no. 107830.

[23] M. Gregory et al., “Commercial optical inter-satellite communication at high data rates,” *Opt. Eng.*, vol. 51, no. 3, 2012, Art. no. 031202.

[24] Y. Guo, Q. Li, Y. Li, N. Zhang, and S. Wang, “Service coordination in the space-air-ground integrated network,” *IEEE Netw.*, vol. 35, no. 5, pp. 168–173, Sep./Oct. 2021.

[25] A. Hagberg and D. Conway, “NetworkX: Network analysis with python,” 2020. [Online]. Available: https://networkx.github.io

[26] R. Hassin, “Approximation schemes for the restricted shortest path problem,” *Math. Operations Res.*, vol. 17, no. 1, pp. 36–42, 1992.

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 13,2024 at 13:29:56 UTC from IEEE Xplore. Restrictions apply.

[27] Y. Henri, “The oneweb satellite system,” in *Handbook of Small Satellites: Technology, Design, Manufacture, Applications, Economics and Regulation*, Berlin, Germany: Springer, 2020, pp. 1–10.

[28] H. Huang, S. Guo, J. Wu, and J. Li, “Service chaining for hybrid network function,” *IEEE Trans. Cloud Comput.*, vol. 7, no. 4, pp. 1082–1094, Fourth Quarter 2019.

[29] M. Huang, W. Liang, Y. Ma, and S. Guo, “Maximizing throughput of delay-sensitive NFV-enabled request admissions via virtualized network function placement,” *IEEE Trans. Cloud Comput.*, vol. 9, no. 4, pp. 1535–1548, Dec. 2021.

[30] X. Jia, T. Lv, F. He, and H. Huang, “Collaborative data downloading by using inter-satellite links in LEO satellite networks,” *IEEE Trans. Wirel. Commun.*, vol. 16, no. 3, pp. 1523–1532, Mar. 2017.

[31] Z. Jia et al., “Joint optimization of VNF deployment and routing in software defined satellite networks,” in *Proc. IEEE 88th Veh. Technol. Conf.*, 2018, pp. 1–5.

[32] Z. Jia, M. Sheng, J. Li, D. Zhou, and Z. Han, “VNF-based service provision in software defined LEO satellite networks,” *IEEE Trans. Wirel. Commun.*, vol. 20, no. 9, pp. 6139–6153, Sep. 2021.

[33] Z. Jia, M. Sheng, J. Li, Y. Zhu, W. Bai, and Z. Han, “Virtual network functions orchestration in software defined LEO small satellite networks,” in *Proc. IEEE Int. Conf. Commun.*, 2020, pp. 1–6.

[34] C. Li, Y. Zhang, X. Hao, and T. Huang, “Jointly optimized request dispatching and service placement for MEC in LEO network,” *China Commun.*, vol. 17, no. 8, pp. 199–208, Aug. 2020.

[35] D. Li, S. Wu, Y. Wang, J. Jiao, and Q. Zhang, “Age-optimal HARQ design for freshness-critical satellite-IoT systems,” *IEEE Internet Things J.*, vol. 7, no. 3, pp. 2066–2076, Mar. 2020.

[36] J. Li, W. Shi, H. Wu, S. Zhang, and X. Shen, “Cost-aware dynamic SFC mapping and scheduling in SDN/NFV-enabled space–air–ground-integrated networks for internet of vehicles,” *IEEE Internet Things J.*, vol. 9, no. 8, pp. 5824–5838, Apr. 2022.

[37] T. Li, X. Zhou, S. Yan, and X. Zhang, “Service function path selection methods for multi-layer satellite networks,” *Peer-to-Peer Netw. Appl.*, vol. 15, no. 5, pp. 2161–2178, 2022.

[38] J. Liu, Y. Shi, Zubair M. Fadlullah, and N. Kato, “Space-air-ground integrated network: A survey,” *IEEE Commun. Surv.s Tut.*, vol. 20, no. 4, pp. 2714–2741, Fourth Quarter 2018.

[39] I. Maity, T. X. Vu, S. Chatzinotas, and M. Minardi, “D-vine: Dynamic virtual network embedding in non-terrestrial networks,” in *Proc. IEEE Wirel. Commun. Netw. Conf.*, 2022, pp. 166–171.

[40] S. Mao, S. He, and J. Wu, “Joint UAV position optimization and resource scheduling in space-air-ground integrated networks with mixed cloud-edge computing,” *IEEE Syst. J.*, vol. 15, no. 3, pp. 3992–4002, Sep. 2021.

[41] S. Poslad, S. E. Middleton, F. Chaves, R. Tao, O. Necmioglu, and U. Bügel, “A semantic IoT early warning system for natural environment crisis management,” *IEEE Trans. Emerg. Top. Comput.*, vol. 3, no. 2, pp. 246–257, Jun. 2015.

[42] X. Qin, T. Ma, Z. Tang, X. Zhang, H. Zhou, and L. Zhao, “Service-aware resource orchestration in ultra-dense LEO satellite-terrestrial integrated 6G: A service function chain approach,” *IEEE Trans. Wirel. Commun.*, vol. 22, no. 9, pp. 6003–6017, Sep. 2023.

[43] J. Sheng et al., “Space-air-ground integrated network development and applications in high-speed railways: A survey,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 10066–10085, Aug. 2022.

[44] J. Shi, J. Wang, H. Huang, L. Shen, J. Zhang, and H. Xu, “Joint optimization of stateful VNF placement and routing scheduling in software-defined networks,” in *Proc. IEEE Int. Conf Parallel Distrib. Process. Appl., Ubiquitous Comput. Commun. Big Data Cloud Comput., Social Comput. Netw., Sustain. Comput. Commun.*, 2018, pp. 9–14.

[45] Z. Song, Y. Hao, Y. Liu, and X. Sun, “Energy-efficient multiaccess edge computing for terrestrial-satellite Internet of Things,” *IEEE Internet Things J.*, vol. 8, no. 18, pp. 14202–14218, Sep. 2021.

[46] Y. Su, Y. Liu, Y. Zhou, J. Yuan, H. Cao, and J. Shi, “Broadband LEO satellite communications: Architectures and key technologies,” *IEEE Wirel. Commun.*, vol. 26, no. 2, pp. 55–61, Apr. 2019.

[47] L. Tan et al., “Speech emotion recognition enhanced traffic efficiency solution for autonomous vehicles in a 5G-enabled space–air–ground integrated intelligent transportation system,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 2830–2842, Mar. 2022.

[48] A. Varasteh et al., “Mobility-aware joint service placement and routing in space-air-ground integrated networks,” in *Proc. IEEE Int. Conf. Commun.*, 2019, pp. 1–7.

[49] G. Wang, S. Zhou, S. Zhang, Z. Niu, and X. Shen, “SFC-based service provisioning for reconfigurable space-air-ground integrated networks,” *IEEE J. Sel. Areas Commun.*, vol. 38, no. 7, pp. 1478–1489, Jul. 2020.

[50] W. Wang, T. Chen, R. Ding, G. Seco-Granados, L. You, and X. Gao, “Location-based timing advance estimation for 5G integrated leo satellite communications,” *IEEE Trans. Veh. Technol.*, vol. 70, no. 6, pp. 6002–6017, Jun. 2021.

[51] N. Waqar, S. A. Hassan, A. Mahmood, K. Dev, D.-T. Do, and M. Gidlund, “Computation offloading and resource allocation in mec-enabled integrated aerial-terrestrial vehicular networks: A reinforcement learning approach,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 21478–21491, Nov. 2022.

[52] Q. Xia, W. Ren, Z. Xu, P. Zhou, W. Xu, and G. Wu, “Learn to optimize: Adaptive VNF provisioning in mobile edge clouds,” in *Proc. IEEE 17th Annu. Int. Conf. Sens. Commun. Netw.*, 2020, pp. 1–9.

[53] 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 Netw.*, vol. 34, no. 3, pp. 224–231, May/Jun. 2020.

[54] Z. Xu, W. Liang, M. Huang, M. Jia, S. Guo, and A. Galis, “Approximation and online algorithms for NFV-enabled multicasting in SDNs,” in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst.*, 2017, pp. 625–634.

[55] Z. Xu, W. Liang, M. Huang, M. Jia, S. Guo, and A. Galis, “Efficient NFV-enabled multicasting in SDNs,” *IEEE Trans. Commun.*, vol. 67, no. 3, pp. 2052–2070, Mar. 2019.

[56] Z. Xu, Z. Zhang, W. Liang, Q. Xia, O. Rana, and G. Wu, “QoS-aware VNF placement and service chaining for IoT applications in multi-tier mobile edge networks,” *ACM Trans. Sensor Netw.*, vol. 16, no. 3, pp. 1–27, 2020.

[57] H. Yao, L. Wang, X. Wang, Z. Lu, and Y. Liu, “The space-terrestrial integrated network: An overview,” *IEEE Commun. Mag.*, vol. 56, no. 9, pp. 178–185, Sep. 2018.

[58] L. You, K.-X. Li, J. Wang, X. Gao, X.-G. Xia, and B. Ottersten, “Massive MIMO transmission for LEO satellite communications,” *IEEE J. Sel. Areas Commun.*, vol. 38, no. 8, pp. 1851–1865, Aug. 2020.

[59] G. Yuan et al., “Fault tolerant placement of stateful VNFs and dynamic fault recovery in cloud networks,” *Comput. Netw.*, vol. 166, 2020, Art. no. 106953.

[60] P. Zhang, Y. Su, J. Wang, C. Jiang, C.-H. Hsu, and S. Shen, “Reinforcement learning assisted bandwidth aware virtual network resource allocation,” *IEEE Trans. Netw. Service Manage.*, vol. 19, no. 4, pp. 4111–4123, Dec. 2022.

[61] P. Zhang, P. Yang, N. Kumar, and M. Guizani, “Space-air-ground integrated network resource allocation based on service function chain,” *IEEE Trans. Veh. Technol*, vol. 71, no. 7, pp. 7730–7738, Jul. 2022.

[62] X. Zhang et al., “Energy-efficient computation peer offloading in satellite edge computing networks,” *IEEE Trans. Mobile Comput.*, early access, pp. 1–15, Apr. 25, 2023, doi: 10.1109/TMC.2023.3269801.

[63] Z. Zhang, W. Zhang, and F.-H. Tseng, “Satellite mobile edge computing: Improving QoS of high-speed satellite-terrestrial networks using edge computing techniques,” *IEEE Netw.*, vol. 33, no. 1, pp. 70–76, Jan./Feb. 2019.

[64] J. Zu, G. Hu, J. Yan, and S. Tang, “A community detection based approach for service function chain online placement in data center network,” *Comput. Commun.*, vol. 169, pp. 168–178, 2021.

Qiufen Xia (Member, IEEE) received the BSc and ME degrees in computer science from the Dalian University of Technology, China, in 2009 and 2012, respectively, and the PhD degree in computer science from the Australian National University, in 2017. She is currently an associate professor with the Dalian University of Technology. Her research interests include mobile edge computing, query evaluation, Big Data analytics, Big Data management in distributed clouds, and network function virtualization.

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 13,2024 at 13:29:56 UTC from IEEE Xplore. Restrictions apply.

Guijie Wang received the BSc degree in software engineering from the Dalian University of Technology, China, in 2021. He is currently working toward the ME degree with the Dalian University of Technology. His research interests include network function virtualization, Internet of Things, and space-air-ground integrated network.

Zichuan Xu (Member, IEEE) received the BSc and ME degrees in computer science from the Dalian University of Technology, China, in 2008 and 2011, and the PhD degree in computer science from the Australian National University, in 2016. From 2016 to 2017, he was a research associate with the Department of Electronic and Electrical Engineering, University College London, U.K.. He is currently a full professor and PhD advisor with the School of Software, Dalian University of Technology. He is also a “Xinghai Scholar’ with the Dalian University of Technology. His research interests include mobile edge computing, serverless computing, network function virtualization, Internet of Things, and algorithm design.

Weifa Liang (Senior Member, IEEE) received the BSc degree in computer science from Wuhan University, China, in 1984, the ME degree in computer science from the University of Science and Technology of China, in 1989, and the PhD degree in computer science from the Australian National University, in 1998. He is a professor with the Department of Computer Science, City University of Hong Kong. Prior to that, he was a professor with the Australian National University. His research interests include design and analysis of energy efficient routing protocols for Internet of Things, mobile edge computing (MEC), network function virtualization (NFV), software-defined networking (SDN), design and analysis of parallel and distributed algorithms, approximation algorithms, combinatorial optimization, and graph theory. He currently serves as an associate editor in the editorial Board of *IEEE Transactions on Communications*.

Zhou Xu received the BSc degree in software engineering from the Dalian University of Technology, China, in 2019. He is currently working toward the PhD degree with the Dalian University of Technology. His research interests include edge computing, network function virtualization, LEO satellite, and UAV communication.

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 13,2024 at 13:29:56 UTC from IEEE Xplore. Restrictions apply.