Fairness-Aware Dynamic VNF Mapping and Scheduling in SDN/NFV-Enabled Satellite Edge Networks

Conference Paper · September 2023
DOI: 10.1109/ICC45041.2023.10279545

6 authors, including:

Haftay Gebreslasie Abreha
University of Luxembourg

Houcine Chougrani
University of Luxembourg

Ilora Maity
University of Luxembourg

Van-Dinh Nguyen
VinUniversity

Fairness-Aware Dynamic VNF Mapping and Scheduling in SDN/NFV-Enabled Satellite Edge Networks

Haftay Gebreslasie Abreha*, Houcine Chougrani*, Ilora Maity*
Van-Dinh Nguyen†, Symeon Chatzinotas*, and Christos Politis‡
*Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg
†College of Engineering and Computer Science, VinUniversity, Hanoi 100000, Vietnam
‡SES S.A., Chateau de Betzdorf, Luxemburg

***Abstract*—Satellite edge computing (SEC) has emerged as a promising technology to deliver network services to remote users. Coupled with software-defined networking (SDN) and network function virtualization (NFV), SEC can provide flexibility, agility, and efficiency when allocating computing and storage resources. However, there still remain a number of technical challenges in terms of fairness and efficiency of the allocation of physical resources in service provisioning, especially in a satellite network with limited resources and dynamic traffic demands. In this paper, we investigate a dynamic virtual network function (VNF) mapping and scheduling in an SDN/NFV-enabled SEC environment to maximize the fairness between competing services in terms of the E2E delay safe margin to enhance the service acceptance rates in the network. We mathematically formulate the VNF mapping and scheduling problem as a nonlinear integer optimization problem, which is NP-hard. In order to effectively solve the problem, this paper proposes a two-stage heuristic dynamic VNF mapping and scheduling algorithm: *i*) *the path selection algorithm* returns all possible paths for a given service request with multiple VNFs, which are sorted in ascending order based on their E2E service delay and executed offline, and *ii*) *the dynamic VNF mapping and scheduling algorithm* performs online dynamic remapping and rescheduling of VNFs. Finally, numerical results are provided to demonstrate that the proposed algorithm offers a higher service acceptance rate, computing resource utilization efficiency, and higher fairness compared to a benchmark scheme.**

I. Introduction

The advent of satellites equipped with micro-servers opens up new opportunities and architectures to provide ubiquitous computing with low latency and global coverage [1], [2]. The in-orbit computing environment lends itself to satellite edge computing (SEC) [3], which reduces latency by harnessing the computing power of nearby satellite nodes. At SEC, satellite nodes act as edge computing devices to process data such as audio or video multimedia processing to reduce service delays and minimize bandwidth costs of uplink/downlink connections. Combining SEC with software-defined networking (SDN) [4] and network function virtualization (VNF) [5] concepts offers a promising paradigm shift for agile and efficient management and orchestration of services via satellites. In this configuration, network functions can be directly deployed in satellites to form the whole network or a part of it with a specific goal, *e.g.*, to reduce the service delay. Due to the programmable nature of network management in SDN/NFV-enabled satellite networks, network functions are deployed in an automated manner, improving the flexibility and agility of network service management and orchestration.

In an SDN/NFV-enabled SEC network architecture, service requests are treated as service function chains (SFC) and processed as specific sequences of VNFs hosted on the SEC nodes. SFC provisioning consists of three steps: VNF node mapping, VNF link mapping, and VNF scheduling [6]. In the VNF node mapping step, VNF instances are placed on an appropriate SEC node for processing. In contrast, in the VNF link mapping, an appropriate physical link is selected to connect two consecutive VNFs (*i.e.*, virtual link that connects two consecutive VNFs) for the SFC request. VNF scheduling determines the appropriate time slot to execute the requested VNF.

Many studies emphasize the deployment of SFCs in satellite-terrestrial integrated networks (STNs) [6]–[13]. Except [6], [12], existing works have focused on VNF mapping without considering the VNF scheduling. The authors in [6] proposed a meta-heuristic algorithm to support online VNF mapping and scheduling. However, they focused on static scenarios without readjusting current scheduling and mapping strategies which may reduce service acceptance rates, especially in high service demands. The authors in [12] presented a method for dynamic scheduling and mapping VNFs to meet the demands of newly arriving service requests. However, none of the existing work considers fairness in terms of the delay margin between services, which is defined as the safe margin of E2E delay to reach the upper bound of the delay requirement for the service. The existing works perform VNF mapping and scheduling sequentially in a *service-by-service* approach. The VNF mapping and scheduling of a service request must be completed before another service can be mapped and scheduled, resulting in their inefficient use of the VNF node and link resources.

To address these challenges, in this paper we propose dynamic VNF mapping and scheduling in an SDN/NFV-enabled SEC environment to maximize E2E delay margin fairness among competing services. Moreover, our proposed solution performs VNF mapping and scheduling at the VNF level of existing services instead of the service-by-service approach. The following are our main contributions:

- We present VNF mapping and scheduling of services in an SDN/NFV-enabled SEC network environment to meet dynamic traffic demands with limited resources.

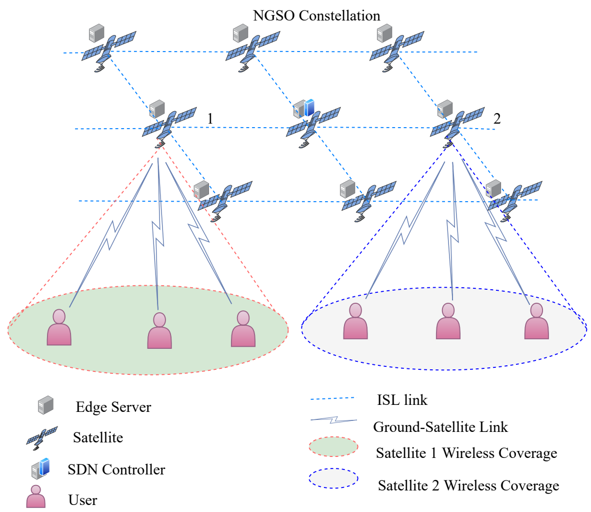


Figure 1: SDN/NFV-enabled SEC network architecture.

- We formulate the VNF mapping and scheduling problem as a nonlinear integer programming (NLIP) problem that maximizes the fairness between competing services in terms of the E2E delay margin.
- We propose a two-stage dynamic SFC mapping and scheduling algorithm to solve the NLIP problem.
- We perform extensive simulations to evaluate the performance of our algorithm by setting the appropriate topology and system parameters that show the effectiveness of the proposed solution.

II. System Model

We consider a general SDN/NFV-enabled SEC network architecture as shown in Figure 1. The network consists of SEC satellite nodes equipped with edge servers in the non-geostationary orbit (NGSO). The SEC nodes have two distinct purposes: processing VNFs and acting as an SDN switch for forwarding SFC traffic from one node to another over the wireless links. In this paper, we assume an SDN controller is placed on one of the SEC nodes and logically centralized to manage the network and orchestrate VNF mapping and scheduling tasks. We assume the well-known +Grid connectivity model [14] as the base topology, where each satellite is connected to two adjacent satellites in the same orbit and two satellites in adjacent orbits (see Figure. 1), using inter-satellite links (ISLs). We also assume that the sources of services (*i.e.*, SFC requests) such as instant messaging and VoIP are remote users that do not have terrestrial network access. In addition, the users communicate directly with the satellites *e.g.*, satellites 1 and 2 via up/downlinks.

A. Substrate Network

In this subsection, we discuss the physical substrate network model to analyze our system model shown in Figure. 1. We model the physical SDN/NFV-enabled SEC network topology as a graph $G = (V, E)$, where V denotes the SEC nodes and E denotes the ISLs between the SEC nodes as well as the ground-satellite wireless links between users and neighboring SEC nodes. We assume that there are $|V|$ SEC nodes in the network, denoted by $V = \{v_1, v_2, .., v_k, ..., v_{|V|}\}$, where each of them is capable of processing the VNFs and storing the forwarding table for routing the SFC traffic. Therefore, a given SEC node $v_k \in V$ is represented by its own processing capacity (CPU) and buffer size, denoted as C_k and β_k, respectively. Moreover, a link connecting nodes v_h and v_k is represented as $e_{h,k} \in E$ and has its own attributes such as bandwidth capacity and transmission latency, denoted as $b_{h,k}$ and $d_{h,k}$ respectively.

In the adopted architecture, the topology of the satellite network is time-dependent, meaning that the availability of links varies with time. Fortunately, the mobility of the NGSO satellites is periodic and predictable [15]. We, therefore, analyze the SDN/NFV-enabled SEC network by dividing the topology into different periodic time windows or snapshots in which the network topology is static. For example, we assume that time is discretized into time slots $T = \{t_1, t_2, ..., t_n, ..., t_N\}$, where T is the satellite's recurrence period, and that the topology of a network is periodic at time T. We therefore define link availability as $\gamma_{h,k}(t_n)$ between nodes v_h and v_k at a snapshot t_n as follows

$$\gamma_{h,k}(t_n) = \begin{cases} 1, & \text{if } \; e_{h,k} \in E \text{ is available at } t_n; \\ 0, & \text{Otherwise.} \end{cases} \quad (1)$$

B. SFC Request Model

Let $SR(t_n)$ be the set of active SFC requests at the snapshot t_n in the network and $\mathrm{SR}_i(t_n)$ is the i^{th} service request at the snapshot represented as $\mathrm{SR}_i(t_n) = \langle src, VF_i, dst, C_i, \beta_i, B_i, D_i \rangle$. The src and dst are the sources and destination nodes where the service SR_i begins and ends respectively. $f_{i,j} \in VF_i$ is the $j-$VNF type of the i^{th} service request. We assume that $c_{i,j} \in C_i$ and $\beta_{i,j} \in \beta_i$ represent the *CPU* and *buffer size* required to process the VNF $f_{i,j}$. B_i and D_i are the minimum E2E bandwidth and the maximum E2E delay tolerated by the service request SR_i, respectively. The node mapping decision of $f_{i,j}$ onto node v_k at snapshot t_n is a binary variable defined:

$$X_k^{i,j}(t_n) = \begin{cases} 1, & f_{i,j} \text{ is mapped onto node } v_k \text{ at } t_n; \\ 0, & \text{Otherwise.} \end{cases} \quad (2)$$

The decision on mapping the virtual link between VNFs $f_{i,j}$ and its preceding VNF represented by $f_{i,j-1}$ to the physical link between nodes v_h and v_k at snapshot, t_n is defined as:

$$Y_{h,k}^{i,j}(t_n) = \begin{cases} 1, & \text{if link } (f_{i,j-1} - f_{i,j}) \text{ is mapped to } e_{h,k}; \\ 0, & \text{Otherwise.} \end{cases} \quad (3)$$

In the VNF scheduling decision, we assume in this work that an SEC node can host multiple VNFs running on virtual machines (VMs) or containers simultaneously so that the node can run numerous VNFs in parallel. We also assume that VMs/containers within each satellite node run VNFs that are uniquely different from those in other VMs/containers within the same satellite node. Therefore, concurrent requests for the same VNF type in a particular VM or container are assigned a priority for scheduling VNFs. In addition, an efficient VNF scheduling scheme is required when the resources on a node are not sufficient to process the requested VNF. We assume that $Z_{j,k}^{i,g}(t_n)$ is a binary variable indicating the scheduling decision on the j−VNF type of service i (*i.e.*, $f_{i,j}$) and the j VNF type of service g (*i.e.*, $f_{g,j}$) at node v_k and can be defined as follows.

$$Z_{j,k}^{i,g}(t_n) = \begin{cases} 1, & \text{if } f_{i,j} \text{ waits till } f_{g,j} \text{ finishes at } v_k;\ \forall v_k \in V \\ 0, & \text{Otherwise.} \end{cases} \quad (4)$$

C. End-to-End (E2E) Service Delay Model

In this work, the E2E service delay consists of the sum of three basic delay components: 1) propagation, 2) processing and 3) waiting delays. Therefore, we formulated the propagation delay of link $e_{h,k}$ at snapshot t_n as $d_{h,k}(t_n)$ and it depends on the distance between the nodes v_h and v_k. Moreover, the processing delay of a VNF $f_{i,j}$ on node v_k is denoted as $\ell_k^{i,j}$. The waiting delay is a cumulative VNF scheduling delay that specifies how long VNFs of service wait at nodes before being processed. Let $A_k^{i,j}$ be the arrival time of the VNF $f_{i,j}$. The VNF $f_{i,j}$ total waiting time at node v_k is denoted by $\tau_k^{i,j}(t_n)$ and can be calculated as:

$$\tau_k^{i,j}(t_n) = \sum_{g=\in SR(t_n)} \left(\ell_k^{g,j} - \left(A_k^{i,j} - A_k^{g,j}\right)\right) \varrho_{j,k}^{i,g} Z_{j,k}^{i,g}(t_n)$$
$$+ \sum_{g\in SR(t_n)} \left(\ell_k^{i,j} - \left(A_k^{g,j} - A_k^{i,j}\right)\right) \varrho_{j,k}^{g,i} Z_{j,k}^{g,i}(t_n), \ \forall v_k \in V. \quad (5)$$

Here $\varrho_{j,k}^{i,g}$ is a binary variable indicating whether the VNF $f_{i,j}$ arrives at node v_k while processing the VNF $f_{g,j}$.

$$\varrho_{j,k}^{i,g} = \begin{cases} 1 & \text{if } A_k^{i,j} - A_k^{g,j} \le \ell_k^{g,j},\ \forall v_k \in V \\ 0, & \text{Otherwise.} \end{cases} \quad (6)$$

Therefore, for the service request SR_i at a snapshot t_n, the total E2E delay is calculated as:

$$d_i(t_n) = \sum_{j=1, e_{h,k}\in E}^{|VF_i|} Y_{h,k}^{i,j}(t_n) d_{h,k}(t_n)$$
$$+ \sum_{j=1}^{|VF_i|} \sum_{k=1}^{|V|} X_k^{i,j}(t_n)\left(\ell_k^{i,j} + \tau_k^{i,j}(t_n)\right). \quad (7)$$

Remark. In general, propagation delay dominates service delay compared to waiting and processing delay. However, the waiting and processing delays can become comparable to the magnitude of the propagation delay when multiple VNFs are mapped on a limited number of nodes.

III. Dynamic VNF Mapping and Scheduling

We assume that SFC requests arrive randomly with specific requirements. In this work, we aim to optimize the dynamic VNF mapping and scheduling strategy to maximize the delay margin fairness among competing services.

A. Problem Formulation

We define the decision variables $X(t_n)$, $Y(t_n)$, and $Z(t_n)$, which specify the VNF node mapping, link mapping, and scheduling at a given snapshot t_n, respectively. The objective function is to maximize the delay margin of the service with a minimum delay margin among the service requests to improve the fairness between the competing services, and is defined using *Max-Min* fairness as follows:

$$\Phi_i(t_n) \triangleq \frac{D_i}{d_i(t_n)} \quad (8a)$$

$$\max_{X(t_n),Y(t_n),Z(t_n)} \min_i \{\Phi_i(t_n)\} \quad (8b)$$

$$\text{s.t.} \quad X_k^{i,j}(t_n) \in \{0,1\},\ \forall f_{i,j} \in \mathrm{SR}_i,\ \forall v_k \in V \quad (8c)$$

$$Y_{h,k}^{i,j}(t_n) \in \{0,1\},\ \forall f_{i,j} \in \mathrm{SR}_i,\ \forall e_{h,k} \in E \quad (8d)$$

$$Z_{j,k}^{i,g}(t_n) \in \{0,1\},\ \forall f_{i,j} \in \mathrm{SR}_i, f_{g,j} \in \mathrm{SR}_g,\ \forall v_k \in V \quad (8e)$$

$$\sum_{v_k\in V} X_k^{i,j}(t_n) = 1,\ \forall f_{i,j} \in \mathrm{SR}_i \quad (8f)$$

$$\sum_{i=1}^{|SR(t_n)|} \sum_{j=1}^{|VF_i|} X_k^{i,j}(t_n) c_{i,j} \le C_k,\ \forall v_k \in V \quad (8g)$$

$$\sum_{i=1}^{|SR(t_n)|} \sum_{j=1}^{|VF_i|} X_k^{i,j}(t_n) \beta_{i,j} \le \beta_k,\ \forall v_k \in V \quad (8h)$$

$$\sum_{e_{h,k}\in E} Y_{h,k}^{i,j}(t_n) \gamma_{h,k}(t_n) \ge 1,\ \forall f_{i,j} \in \mathrm{SR}_i \quad (8i)$$

$$\sum_{i=1}^{|SR(t_n)|} \sum_{j=1}^{|VF_i|} Y_{h,k}^{i,j}(t_n) B_i \gamma_{h,k}(t_n) \le b_{h,k}(t_n),\ \forall e_{h,k} \quad (8j)$$

$$Z_{j,k}^{i,g}(t_n) + Z_{j,k}^{g,i}(t_n) \le 1, \forall f_{i,j} \in \mathrm{SR}_i, f_{g,j} \in \mathrm{SR}_g, \forall v_k. \quad (8k)$$

Where $\Phi_i(t_n)$ shown in Equation (8a) is the safe delay margin of the service request SR_i. The objective function depicted in Equation (8b) is to maximize the delay margin of service with minimum delay margin. Constraints (8c), (8d), and (8e) indicate binary decision variables. Constraint (8f) states that each service's VNF is processed by exactly one node. Furthermore, constraints defined in (8g) and (8h) specify that a VNF can only be mapped to a node with sufficient storage and processing capacity. Constraint (8i) explains that each virtual link can be mapped to multiple physical links. Constraint (8j) guarantees that a virtual link can only be mapped onto links with sufficient bandwidth for VNFs to transmit. Finally, the VNF scheduling constraint given in (8k) specifies that no more than one VNF of the same type can be processed simultaneously at one node We classify constraints that must be met for successful SFC implementation into two categories, *namely* VNF mapping constraints and VNF scheduling constraints. Constraints (8f)-(8j) and (8k) are imposed to ensure the successful VNF mapping and VNF scheduling, respectively.

Note that (8) is a nonlinear integer programming (NLIP) problem with integer decision variables $X_k^{i,j}(t_n)$, $Y_{h,k}^{i,j}(t_n)$, and $Z_{j,k}^{i,g}(t_n)$. The VNF mapping and scheduling problems are NP-hard problems that are not solved in polynomial time. Heuristic algorithms are commonly used to solve these problems [9]–[11], [16]–[18]. Accordingly, we also propose a heuristic algorithm to solve the problem.

B. The Proposed Algorithm for Dynamic VNF Mapping and Scheduling

In this subsection, we discuss the dynamic VNF mapping and scheduling algorithm proposed to solve the NLIP problem, which consists of two steps: *1)* path selection algorithm and *2)* dynamic VNF mapping and scheduling algorithm.

1) Path Selection Algorithm

The SDN controller has a complete view of the satellite network topology. Furthermore, we assume that the service type the SEC network provides is known. Thus, the controller can compute all possible paths offline for all service types that can be delivered by the network. The path selection algorithm provides the sets of sorted paths based on the E2E delay of the service request, $\mathtt{SR}_i$ at a snapshot, t_n denoted as $\mathbf{P}_i(t_n)$. The shortest path in the set $\mathbf{P}_i(t_n)$ that satisfies the conditions listed in (8c)-(8i) is taken as initial mapping and scheduling strategy of service $\mathtt{SR}_i$.

Algorithm 1 DYNAMIC VNF MAPPING AND SCHEDULING

Input: SFC request and $\mathtt{SR}_i, \forall i$ at snapshot t_n
Output: $\boldsymbol{\pi}$ (i.e. Global VNF mapping and scheduling strategy)
1: Extract all VNF components of $\mathtt{SR}_i$ as VNF_i;
2: $\pi_i \leftarrow$ Initialize VNF mapping and scheduling strategy for $\mathtt{SR}_i$;
3: **for** each VNF components of service $\mathtt{SR}_i$, $f_{i,j} \in \mathrm{VNF}_i$ **do**
4: **for** each node, $k \in V$ in the selected path **do**
5: $A_k^{i,j} \leftarrow$ compute arrival time of $f_{i,j}$ at node v_k;
6: **end for**
7: **end for**
8: **for** each $\mathtt{SR}_g \in SR(t_n)$ and $g \neq i$ **do**
9: **for** each node, $v_k \in V$ part of the selected path **do**
10: $\tau_k^{i,j} \leftarrow$ waiting time of $f_{i,j}$ at node v_k using (5);
11: $\tau_k^{g,j} \leftarrow$ waiting time of $f_{g,j}$ at node v_k using (5);
12: **end for**
13: **while** *stopConditionisNotMet()* **do**
14: Compute E2E delay, d^i using (7);
15: Compute E2E delay, d^g using (7);
16: Compute the delay margin, $\boldsymbol{\phi}(t_n)$ using (8a) for all $SR(t_n)$;
17: **if** $\min(\boldsymbol{\phi}(t_n)) \geq \phi_{prev}$ **and** $d^g \leq D^g$ **then**
18: $\phi_{prev} \leftarrow \min(\boldsymbol{\phi}(t_n))$;
19: $\pi_i \leftarrow \{X_i(t_n), Y_i(t_n), Z_i(t_n)\}$: Update the VNF mapping and scheduling for $\mathtt{SR}_i$;
20: $\pi_g \leftarrow \{X_g(t_n), Y_g(t_n), Z_g(t_n)\}$: Update the VNF mapping and scheduling for $\mathtt{SR}_g$;
21: **end if**
22: Exclude the current and select next best path in $\mathbf{P}_i(t_n)$;
23: Perform VNF mapping and scheduling on the path;
24: **end while**
25: **end for**
26: **return** $\boldsymbol{\pi} \leftarrow \{\pi_1, \cdots, \pi_{|SR(t_n)|}\}$

2) Dynamic VNF Mapping and Scheduling Algorithm

Algorithm 1 describes the details of the online dynamic VNF mapping and scheduling algorithm. When a new service request $\mathtt{SR}_i$ arrives at an SEC node at a given snapshot t_n, the proposed algorithm implemented in the SDN controller first extracts the VNF components of the service as shown in line 1. Then, the service VNF mapping and scheduling strategy of $\mathtt{SR}_i$ denoted by π_i[1] is initialized by selecting the shortest path from $\mathbf{P}_i(t_n)$ using the *path selection algorithm* that satisfies the constraints described at in (8c) - (8i) as shown in line 2. The VNF arrival time[2] is calculated for all the VNF components of the services as shown in lines 3 – 6. Next, each selected node must be checked for conflicts of interest. A conflict of interest occurs when VNF requests arrive at a node that does not have enough resources to execute them simultaneously. With an efficient VNF scheduling scheme, the SFC request can be processed on the node while ensuring their requirements are met. To this end, the waiting time for the VNF components of all services $\mathtt{SR}(t_n)$ in the network is calculated according to a first-come-first-served (FCFS) scheme, where the VNF that arrives first at the specified node is processed before the one that arrives later, as shown in lines 8 – 12. The VNF waiting time is introduced to mitigate conflicts of interest at the VNF level by applying the FCFS scheme regardless of the arrival time of services. Therefore, the waiting time is imposed on all services actively available in the network; since our algorithm handles scheduling at the VNF level and not at the service level. In other words, the proposed VNF scheduling is done in parallel for all services available in the network based on the arrival time of the VNFs, rather than scheduling sequentially for each service when a new service request arrives at the network. As shown in line 10, the waiting time is calculated for the current request and another competing service that shares the same node with the newly arriving service, $\mathtt{SR}_i$ as shown in line 11.

Next, the E2E service delay of the current service request $\mathtt{SR}_i$ (i.e., d^i), the concurrent service $\mathtt{SR}_g$ (i.e., d^g), and the delay margin vector, $\boldsymbol{\phi}(t_n)$ which consists of all active services in the network, $SR(t_n)$ using (8a) as shown in lines 14, 15, and 16 respectively, are computed. The E2E service of each service in the network is checked if they do not violate the delay demand requirements. In addition, since our goal is to maximize the objective function defined in the objective 8b, the algorithm checks if the current minimum delay margin (*i.e.*, $\min(\boldsymbol{\phi}(t_n))$) is larger than the previous value, (*i.e.*, ϕ_{pre}) as shown in line 17. When all service delays demands are satisfied and if the minimum delay margin of the current value is greater than the previous value, the VNF mapping and scheduling strategies π_i and π_g of the current service request $\mathtt{SR}_i$ and the competing service $\mathtt{SR}_g$, respectively, are updated as shown in lines 19 and 20.

Since the goal is to find the VNF mapping and scheduling strategy that maximizes the E2E delay margin fairness of services, the best strategy with a maximum of the minimum service delay margin is searched until the stop condition (*i.e.*, *stopConditionisNotMet()*) is satisfied by selecting next shortest path from $\mathbf{P}_i(t_n)$ and performing the corresponding VNF mapping and scheduling, as shown in lines 22 and 23 at every iteration. Iteration stops when: *1)* the number of iterations exceeds the maximum, *2)* the objective function value (*i.e.*, $\min(\boldsymbol{\phi}(t_n))$) remains unchanged for certain it-

[1] $\pi_i \triangleq \{X_i(t_n), Y_i(t_n), Z_i(t_n)\}$ indicates the VNF node, link mapping and scheduling strategy of service $\mathtt{SR}_i$ at snapshot t_n.

[2] VNF arrival time is defined as the time at which a VNF requests a corresponding node for processing.

erations, or *3)* there is no candidate path in $\mathbf{P}_i(t_n)$ that satisfies the constraints. Finally, it updates the global VNF mapping and scheduling strategy, $\boldsymbol{\pi}^3$ with the maximum of the minimum service delay margin. A service gets rejected when there is no available path among the candidate paths in $\mathbf{P}_i(t_n)$ that satisfies the constraints. The proposed algorithm depicted in Algorithm 1 computational complexity is defined as $O(I_{tr} \cdot |V| \cdot |SR(t_n|^2)$. Where $|V|$ and $|SR(t_n)|$ are the total number of SEC nodes and competing services at snapshot, t_n in the network respectively. Furthermore, I_{tr} represents the maximum number of iterations to find the optimum solution. As a result, the time complexity of the proposed algorithm is a polynomial function of the network size and service requests which is computationally efficient.

IV. Performance Evaluation

A. Simulation Settings

In our simulations, we use a satellite network with a constellation of $K = 9$ MEO satellites with 3 orbital planes arranged in a +Grid network topology, where each orbital plane has 3 satellites as shown in Figure. 1. We use the $M/M/1$ queuing model to model the interarrival time between SFC requests. The source and destination of the service are randomly selected from users on the ground. The source user and the target user can be the same, which means that users can request a service and get the service back. The other simulation parameters and values used are summarized in Table I.

Table I: Simulation Parameters

Parameters	**Value**
Number of SEC (K)	9
Service Arrival Rate(λ)	$[0.05 - 0.20/ms]$ [12]
Processing delay of SEC node	$[5 - 10]$ ms [12]
ISL propagation delay	$[20 - 40]$ ms [13]
Uplink/downlink propagation delay	$[29 - 43]$ ms [19]
Maximum tolerable E2E service delay	$[150 - 200]$ ms
Number of VNFs per an SFC request	$[3 - 5]$ [13]
Number of VNFs per an SEC node	$[1 - 5]$ [13]
Processing capacity of an SEC node	96 vCPUs [10]
Storage capacity of an SEC node	112 GB [10]
Processing units required by a VNF	$[2 - 4]$ vCPU [10]
Storage units required by a VNF	$[4 - 8]$ GB [10]
capacity of physical ISL	1 Gbps [10]
capacity of physical uplink/downlink	1 Gbps [10]
capacity required by a virtual link	$[10 - 20]$ Mbps [10]

B. Benchmark Schemes

We consider TS_MAPSCH [12] as a benchmark scheme for performance evaluation. TS_MAPSCH completes VNF mapping and scheduling of service requests on a *service-by-service* basis, where VNF mapping and scheduling of one service must be completed before another service can begin. TS_MAPSCH completes the VNF mapping and scheduling strategy by selecting a VNF node with the earliest node to execute the requested VNF among candidate nodes, sequentially for all VNF components of the service.

[3] $\boldsymbol{\pi} \triangleq \{\pi_i, \cdots, \pi_{|SR(t_n)|}\}$

Moreover, the remapping and rescheduling are triggered only when the E2E service delay demand is violated.

C. Results and Discussion

1) Service Acceptance Rate

The service acceptance rate is defined as the ratio between the total number of SFC requests that satisfy the E2E service delay request and the total number of SFC requests. As shown in Figure. 2a, the proposed solution has a higher service acceptance rate compared to the benchmark (*i.e.* TS_MAPSCH). At a service rate of $\lambda = 0.05$, we observe a 3% gap between the proposed algorithm and the benchmark one. This even goes higher when the service rate is high (e.g. at $\lambda = 0.2$ a gap of 10%). There are two main reasons for this performance improvement. *i)* Unlike the benchmark, our proposed algorithm considers the fairness of the delay margin between competing services, which increases the acceptance rate by balancing the delay margin between service requests as described in subsection IV-C2. *ii)* Efficient resource utilization in our proposed algorithm. In fact, our proposed solution performs the VNF mapping and scheduling at the VNF level, which schedules the VNFs of the existing services in the network taking into account the waiting time while deploying the current service request subsection IV-C3. In contrast, the benchmark algorithm performs the VNF mapping and scheduling of services sequentially in a *service-by-service* approach.

As satellite networks are resource constrained, the acceptance rate of services decreases when the arrival rate increases. In contrast to the benchmark, our solution strives to balance the service delay margin between competing requests to accept more services even at a higher arrival rate. Due to this, the proposed solution outperforms the benchmark sufficiently at higher service arrival rates.

2) E2E Service Delay Margin Fairness

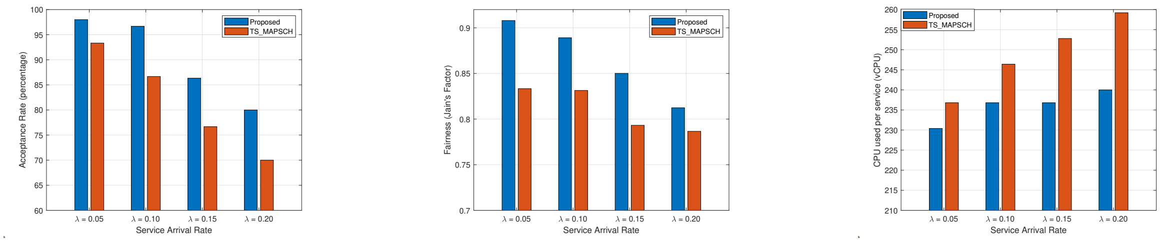
Fairness is described in terms of service delay margin (*i.e.,* $\Phi_i(t_n)$) and defined using Jain's index [20] normalized with their corresponding delay demand, D_i so that each request has the same weight as outlined below.

$$\mathrm{J} = \frac{\left(\sum_{i=1}^{|SR(t_n)|} \Phi_i(t_n)\right)^2}{\left|SR(t_n)\right| \sum_{i=1}^{|SR(t_n)|} \Phi_i^2(t_n)} \tag{9}$$

As shown in Figure. 2b, the proposed algorithm outperforms TS_MAPSCH for all service arrival rates. Thus, the proposed solution outperforms the benchmark algorithm by about a 9% and 3% fairness gap for of $\lambda = 0.05$ and $\lambda = 0.2$, respectively. This is because, unlike the benchmark algorithm, the proposed solution jointly maximizes the delay margin of all competing services, leading to a balanced margin between them. As expected, fairness is inversely proportional to the arrival rate of the services. This is because, with more service requests and limited resources, the degree of freedom for fairness decreases.

3) Average Computing Resource Used Per Service

The computing resource used per service is defined as the total amount of computing resources (*i.e.,* CPUs)



(a) Service acceptance rate.

(b) Jain's fairness index.

(c) Average computing resource used per service.

Figure 2: Performance evaluation and comparison.

required for processing the VNF components to complete a corresponding service request. As shown in Figure. 2c, the proposed algorithm uses fewer computing resources (*i.e.,* virtual CPUs (vCPUs)) than `TS_MAPSCH`. As previously explained, the proposed solution utilizes resources more efficiently, resulting in fewer VNF nodes and links. With the VNF-level mapping and scheduling strategy (*i.e.,* fine-grained) instead of the *service-by-service* scheme, the proposed algorithm ensures better resource utilization than the benchmark, which consumes fewer vCPUs.

V. Conclusion

In this paper, we have presented a new dynamic VNF mapping and scheduling solution in an SDN/NFV-enabled SEC network. We formulated the dynamic VNF mapping and scheduling problem as NLIP with limited satellite resources. We propose a two-stage heuristic algorithm to maximize the fairness between competing services in terms of the E2E delay safe margin. The first stage is a path selection algorithm for VNF mapping and scheduling initialization, which is carried out offline. In addition, the second stage includes dynamic online VNF mapping and scheduling that enables dynamic remapping and rescheduling. The simulation results show that the proposed algorithm has a higher service acceptance rate, more fairness, and uses fewer vCPUs than the benchmark algorithm. In future work, we plan to investigate SFC routing in SEC networks with time-varying topology.

Acknowledgment

This work was supported by the Luxembourg National Research Fund (FNR) under the project INSTRUCT (IPBG19/14016225/ INSTRUCT). We thank Dr. Tedros Salih, who assisted in the optimization problem formulation.

References

[1] D. Bhattacherjee, S. Kassing, M. Licciardello, and A. Singla, "In-orbit computing: An outlandish thought experiment?" in *Proc. 19th ACM Works. Hot Topics in Networks*, 2020, pp. 197–204.

[2] B. Denby and B. Lucia, "Orbital edge computing: Nanosatellite constellations as a new class of computer system," in *Proc. Twenty-Fifth Inter. Conf. Architec. Support for Prog. Lang. and Oper. Sys.*, 2020, pp. 939–954.

[3] Z. Zhang *et al.*, "Satellite mobile edge computing: Improving QoS of high-speed satellite-terrestrial networks using edge computing techniques," *IEEE Network*, vol. 33, no. 1, pp. 70–76, 2019.

[4] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, 2014.

[5] R. Mijumbi *et al.*, "Network function virtualization: State-of-the-art and research challenges," *IEEE Commun. Surv. & Tut.*, vol. 18, no. 1, pp. 236–262, 2015.

[6] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and S. Davy, "Design and evaluation of algorithms for mapping and scheduling of virtual network functions," in *Proc. IEEE Conf. Network Soft. (NetSoft)*, 2015, pp. 1–9.

[7] 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, 2020.

[8] Q. Li, S. Wang, X. Ma, Q. Sun, H. Wang, S. Cao, and F. Yang, "Service coverage for satellite edge computing," *IEEE Internet of Things J.*, vol. 9, no. 1, pp. 695–705, 2021.

[9] X. Gao, R. Liu, and A. Kaushik, "Virtual network function placement in satellite edge computing with a potential game approach," *IEEE Trans. Net. and Service Management*, 2022.

[10] X. Gao, R. Liu, A. Kaushik, and H. Zhang, "Dynamic resource allocation for virtual network function placement in satellite edge clouds," *IEEE Trans. Netw. Science and Engineering*, 2022.

[11] X. Gao, R. Liu, and A. Kaushik, "Virtual network function placement in satellite edge computing with a potential game approach," *IEEE Trans. Network and Service Manag.*, 2022.

[12] 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 of Things J.*, vol. 9, no. 8, pp. 5824–5838, 2021.

[13] I. Maity, T. X. Vu, S. Chatzinotas, and M. Minardi, "D-vine: Dynamic virtual network embedding in non-terrestrial networks," in *IEEE Wire. Commun. and Net. Conf. (WCNC)*. IEEE, 2022, pp. 166–171.

[14] S. Innovation and E. D. Canada. SpaceX FCC update. (2018). [Online]. Available: https://www.ic.gc.ca/eic/site/smtgst.nsf/vwapj/SLPB-005-18-SpaceX-attachment2.pd

[15] J. Li, H. Lu, K. Xue, and Y. Zhang, "Temporal netgrid model-based dynamic routing in large-scale small satellite networks," *IEEE Trans. Veh. Tech.*, vol. 68, no. 6, pp. 6009–6021, 2019.

[16] F. Bari, S. R. Chowdhury, R. Ahmed, R. Boutaba, and O. C. M. B. Duarte, "Orchestrating virtualized network functions," *IEEE Trans. Net. and Service Manag.*, vol. 13, no. 4, pp. 725–739, 2016.

[17] R. Cohen, L. Lewin-Eytan, J. S. Naor, and D. Raz, "Near optimal placement of virtual network functions," in *IEEE Conf. Comp. Commun. (INFOCOM)*, 2015, pp. 1346–1354.

[18] M. Xia, M. Shirazipour, Y. Zhang, H. Green, and A. Takacs, "Network function placement for NFV chaining in packet/optical datacenters," *J. Ligh. Tech.*, vol. 33, no. 8, pp. 1565–1570, 2015.

[19] A. S. . D. M. E. Software. Ansys | Engineering Simulation Software. (25 Sept. 2022). [Online]. Available: https://www.ansys.com/products/missions/ansys-stk

[20] R. K. Jain, D.-M. W. Chiu, W. R. Hawe *et al.*, "A quantitative measure of fairness and discrimination," *Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA*, vol. 21, 1984.