An Aggressive Migration Strategy for Service Function Chaining in the Core Cloud

Haoxian Feng, Zhaogang Shu, Tarik Taleb, Yuantao Wang, Zhiwei Liu

***Abstract*—Service Function Chaining (SFC) is regarded as an important concept for next-generation communication networks because it can flexibly tackle diverse usage scenarios. Due to SFC requests' life-cycle and resource adjustment, the distribution of the remaining physical resources may become unbalanced, which brings negative effects to subsequent SFC requests as well as network operators. In this paper, we investigate the network SFC migration problem in the core cloud under the premise of considering the migration cost and the balance of physical resource distribution. We first model the SFC migration problem as an integer linear program and propose an aggressive migration strategy that can effectively reduce the imbalance of physical resource distribution. Then, we employ two state-of-the-art heuristics to allocate resources for subsequent SFC requests. The simulation results show that migrating SFC requests in the initial service queue can bring favorable feedback to subsequent requests as well as network operators. Compared to the conservative migration strategy, our proposed migration strategy can mitigate the imbalance of physical resource distribution more effectively, and thus the acceptance ratio of subsequent SFC requests, physical resources utilization, and the long-term profit of network operators can be further improved.**

***Index Terms*—Network softwarization, Network function virtualization, Service function chain migration, and resource allocation.**

I. Introduction

As technology evolves, a range of new vertical use cases emerges [1], such as online education, virtual reality, and autonomous manufacturing. However, today's mobile communication networks employ a one-size-fits-all approach to providing services, regardless of the diverging requirements of vertical services. In addition, some traditional network functions, such as firewalls, load balancing, and deep packet inspection, are mostly carried by specific physical devices. Therefore, in order to meet the ever-increasing performance demands of users, network operators have to spend expensive capital to maintain old equipment as well as purchase new equipment. Consequently, it is necessary to update the network architecture with the ability to address diverse application demands and decrease network operators' operation expenses.

Software Defined Networks (SDN) [2] and Network Function Virtualization (NFV) [3] have emerged as promising approaches to address the above limitations. As SDN decouples the control plane and data plane, it can manage the network in a centralized, flexible way. NFV utilizes virtualization technology to separate specific network functions from dedicated devices to general-purpose commodity hardware. Thus Virtual Network Functions (VNFs) can be placed dynamically at appropriate locations on the network to provide services to the users. Meanwhile, with the rapid development of NFV, VNFs can not only scale dynamically under various traffic conditions to avoid performance degradation and guarantee users' Quality of Experience (QoE) [4], [5], but can be also deployed in clusters to overcome the shortcomings of single-point failure and scalability [6].

Based on SDN/NFV technologies, Service Function Chaining (SFC) [7], standardized by the Internet Engineering Task Force (IETF), is regarded as an important networking concept to provide users with flexible services. Typically, SFC comprises a sequence of VNFs and the traffic needs to be steered to traverse these VNFs in a predefined order [8], [9]. As it is essential to deploy SFC onto the physical network, at present, there has been some research on the SFC deployment problem [10]–[13] to focus on how to reduce network providers' operational costs, increase resource utilization rate, and guarantee users' Quality of Service (QoS). When a SFC request is received, the management and orchestration (MANO) layer of the network system allocates resources and makes configurations for it, and also takes responsibility for recycling resources after the life-cycle of a request [14]. Furthermore, to avoid QoE and Service Level Agreement (SLA) violations caused by traffic load variations [15], [16], MANO should dynamically adjust the allocated resources of SFC requests [17], [18]. In addition, to save the allocated resources [19], and adapt to the users' mobility [20], MANO sometimes even has to migrate some VNFs of a request. However, the above adjustment process may result in the unbalanced distribution of physical resources. Different from previous research work [18]–[22], our work mainly answers the following two questions. *A: Will uneven distribution of physical resources adversely affect subsequent SFC requests and network operators? B: Can we reduce the potential negative impact of the uneven distribution of physical resources by migrating SFC requests in the initial service queue?*

Due to resource adjustment and the departures of some SFC

This research work was supported in part by the Fujian Province Natural Science Foundation of China under Agreement 2020J01574 and the Industry-university-research Innovation Fund of China under Agreement 2021FNA05003. This work was also supported by the CHARITY project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101016509. It is also supported in part by the Academy of Finland 6Genesis project under Grant No. 318927 and IDEA-MILL with grant number 33593. (*Corresponding authors*: *Tarik Taleb* and *Zhaogang Shu*)

Haoxian Feng, Zhaogang Shu, Yuantao Wang and Zhiwei Liu are with the computer and information college, Fujian Agriculture and Forestry University, Fuzhou, China. Email: fenghx@fafu.edu.cn; zgshu@fafu.edu.cn; ytwang@fafu.edu.cn; liuzhiwei@fafu.edu.cn

Tarik Taleb is with the Center of Wireless Communications, The University of Oulu, Finland. Email: tarik.taleb@oulu.fi.

requests, after a period of time, some VNF-enabled nodes and links of the substrate network may be overloaded, which may bring negative effects on the subsequent SFC requests and network operators (e.g., lower request acceptance ratio, lower long-term profit, ect.). Let us use the example in Fig. 1 to illustrate the SFC migration problem in the core cloud. In the initial service queue, there are a total of three SFC requests, and the resource requirements of SFC request 1 and SFC request 2 are small. Since it is easier to find viable candidate resource allocation solutions for requests with small resource requirements in this case, MANO can consider migrating SFC request 1 or SFC request 2. We take the migration of SFC request 2 as an example. As shown in Fig. 1, SFC request 2's transmission path can be migrated from (S1 → V1 → V2 → V4 → S5) to (S1 → V1 → V3 → S4 → V4 → S5), and its VNF1, VNF2 can be migrated from V2, V4 to V1, V3, respectively. Although SFC migration brings costs, it can make the distribution of physical resources more balanced, which may have a positive feedback on subsequent SFC requests and network operators.

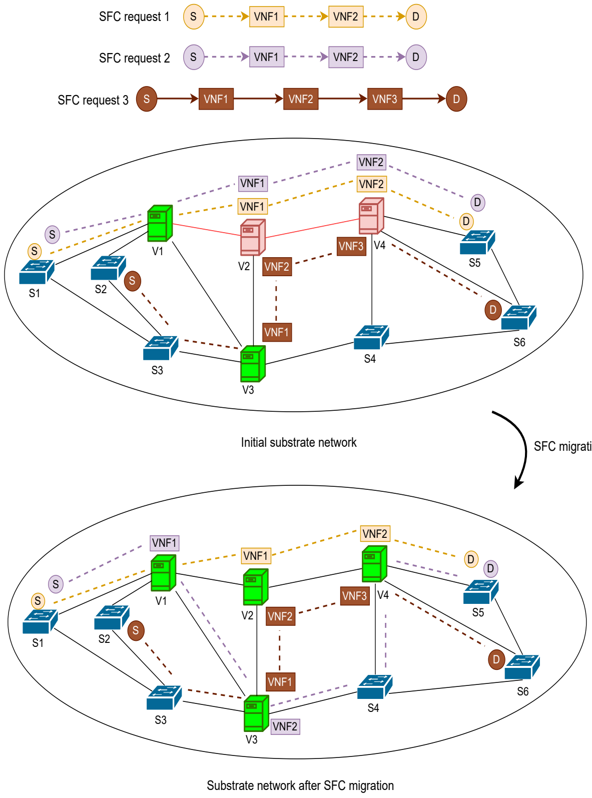


Fig. 1. Illustration of network SFC migration

In this paper, we investigate the SFC migration problem caused by the imbalanced distribution of physical resources in the core cloud. We first use the standard deviation to represent the physical resource distribution of the substrate network, then, jointly considering the migration cost and some constraints (e.g., resource limitation, latency constraint, ect), we formulate the SFC migration problem as an Integer Linear Programming (ILP) problem. Inspired by some relevant work [20], [22], we design a conservative migration strategy to mitigate the imbalance of physical resource distribution. However, the above-mentioned work focused too much on the migration cost. Consequently, only some VNFs (i.e., sometimes only the last VNF of the SFC) are migrated. Therefore, it may not effectively balance the remaining physical resources. To overcome this shortcoming, we propose an aggressive migration strategy to migrate the whole SFC. After that, we employ two state-of-the-art heuristics methods, BestFit [23], [24], and CN [25] to allocate resources for the subsequent SFC requests, and observe the changes in some metrics, such as the acceptance ratio of subsequent SFC requests and the operator's long-term profit, before and after SFC migration. The main contributions of this paper are summarized as follows:

- We formulate the SFC migration problem as an ILP with the aim of minimizing the standard deviation of physical resource distribution as well as the migration cost.
- We propose an aggressive SFC migration strategy that prioritizes mitigating imbalances in resource distribution by migrating the total SFC. Then, we employ two state-of-the-art heuristics to allocate resources for the subsequent SFC requests.
- Simulation results reveal that our proposed migration strategy can alleviate the imbalance of initial network resource distribution by 14.991% compared with the benchmark strategy. For the different resource allocation heuristics, after SFC migration, the acceptance ratio of subsequent SFC requests increased by 11.75% and 6.84% on average, respectively, and the long-term profit of network operators increased by 9.49% and 7.84% on average, respectively.

The remainder of this paper is organized as follows- Section II reviews some related work. Section III formulates the target problem. Section IV introduces our proposed heuristics. Section V presents the obtained numerical results. Finally, the paper concludes in Section VI.

II. Related Work

Network slicing and SFC migration are two hot research topics [26]. In some application scenarios, e.g., automated driving, the location of a user's terminal changes as per the mobility of the user. Taleb *et al.* [27] introduced the Follow-Me Cloud (FMC) concept and proposed its framework. Then, they applied a Markov Decision Process (MDP) based method to optimize migration decisions [28]. Aissioui *et al.* [29] applied their Follow-Me edge Cloud solution in an automated driving use case. When migrating the VNFs of SFC, operators also need to consider reducing the migration time. Addad *et al.* carried out related research work [30], [31] on this. For applications depending on synchronization, they designed, proposed and evaluated four SFC migration patterns [30], and they found no clear winner in their presented patterns. Further, they designed a shared file system-based method [31] to decrease both the migration time and the service downtime.

Although some studies aim to support the mobility of services, their ultimate optimization goals are different. In the hybrid environment of the core cloud and edge cloud, Mada *et al.* [19] formulated the SFC migration problem as an ILP and used the Gurobi optimizer to solve it. In their work, researchers focused on how to minimize the allocated resources to the deployed services but ignored migration cost. Due to the high computation cost of the Gurobi optimizer, their method is infeasible in larger networks. Zhao *et al.* [20] investigated the problem of SFC migration caused by user movement in cloud-fog computing environments. They also formulated the SFC migration problem as an ILP and proposed two heuristic methods to solve it. Considering migration cost, their methods migrate some VNFs of SFC. However, they ignored the life cycles of SFCs and the distribution of the remaining physical resources.

Some researchers employed Artificial Intelligence (AI) methods to solve the slice and SFC migration problem. Addad *et al.* [21] designed, modeled, and evaluated two Deep Reinforcement Learning-based algorithms to allocate bandwidth resources for SFCs during migration. Their experiment results showed that a Deep Deterministic Policy Gradient (DDPG) had better performance than a Deep Q-Network (DQN). However, their proposed AI agent can only detect the required bandwidth for a given workflow, and their work did not consider migration cost. Wei *et al.* [18] investigated the problem of slice reconfiguration caused by fluctuations in user resource requirements in core cloud environments. They formulated the reconfiguration problem as a MDP problem and proposed a discrete Branching Dueling Q-network to solve it. Although their method took the reconfiguration cost into account, their action set was designed for an individual slice, and the trained network model was not universal for SFCs with different resource requirements. For the above DRL-based solutions, once the environment or action set changes, e.g., network topology, SFC requirements, etc., they must retrain the AI agent [32]. As the training of deep network models generally requires high computational and time overhead, these methods have certain limitations. In addition, none of the above research work considers the balance of physical resource distribution after migration or reconfiguration.

Few researchers have considered migrating the VNFs running on overloaded nodes. Zhang *et al.* [22] first proposed an adaptive interference-aware method to allocate resources for slice requests, which can handle VNF interference [33] caused by resource contention. Then, to further improve the long-term total profit of slices and to cope with the migration cost, they proposed a lazy migration strategy that migrates the last VNF of slices to optimize physical resource distribution, so that the network operator can accept slice requests that should be rejected. Although their method can alleviate the burden of high-load nodes to a certain extent, it does not consider the balance of resource distribution from a global perspective.

At present, most of the research on slice and SFC migration is related to supporting users' mobility, and a small part [22] to optimizing global resources. However, to the best of our knowledge, there is no prior work that considers the impact of initial physical resource distribution on subsequent SFC requests and network operators' long-term profit, which is the focus of our work in this paper. A comparison of related works is given in Table **??**.

III. Problem statement and formulation

In this paper, we study the SFC migration problem for SFC requests in the initial service queue. In a physical network, operators can allocate resources for users based on their specific QoS requirements. The QoS requirements and life-cycles of different requests are diverse. Furthermore, during their life cycles, MANO may dynamically adjust the allocated resources according to changes in demand. The above factors cause the resources allocation solutions for the SFC requests in the service queue to become unstable, resulting in the uneven distribution of physical resources.

In this paper, we consider the following scenario. At the initial moment, there are some permanent SFC requests in the service queue, these requests occupy more physical resources than normal requests, and the allocation solutions of their resources are random. MANO needs to migrate these requests under the premise of some constraints (e.g., the association of VNF instances, bandwidth demand, etc.) to alleviate the uneven distribution of physical resources.

Since the essence of the SFC migration problem is to calculate new resource allocation solutions for the requests in the service queue, we first describe the substrate network and SFC request models and then propose a mathematical formulation of the SFC migration problem. For ease of reference, the notations used in this paper are summarized in Table II.

A. Substrate Network

Similar to some related work, we define the substrate network as a weighted directed graph $G = (N, E)$, where N and E denote the sets of physical nodes and links, respectively. Typically, a substrate network is composed of VNF-enabled nodes N_V and common nodes N_F, so the set of physical nodes can be further denoted as $N = N_V \cup N_F$. VNF-enabled nodes can provide certain types of VNFs, such as Network Address Translation (NAT) and Firewall; while common nodes can be only used for packet forwarding. Note that the forwarding function can be seen as a special network function [18] and its resource consumption is relatively small, so the resource consumption caused by the forwarding function is ignored in our work.

We indicate the total computational resource capacity of the VNF-enabled node $i \in N_V$ as C_i, while other resources (e.g., memory resources, storage resources) are sufficient. Each physical link $(i, j) \in E$ has the limited bandwidth of $B_{i,j}$, and the propagation delay of a physical link (i, j) is denoted as $\tau_{i,j}$.

B. SFC Request

In this paper, we define a SFC request r as a linear chain, and it can be denoted as $r = (s_r, d_r, \mathcal{F}_r, \tau_r, c_r, b_r)$,

TABLE I
COMPARISON OF RELATED WORKS

Literature	Applicable scenario	Motivation for migration	Migration target	Methodology	Consider migration cost	Consider life-cycle of services	Computing costs	migration magnitude
Mada et al. [19]	Hybrid environment of core cloud and edge cloud	Users' mobility	save the allocated resources	ILP formulation with Gurobi optimizer	No	Yes	Massive with large networks	Not clear
Zhao et al. [20]	Cloud-fog computing environment	Users' mobility	Guarantee users' QoE	ILP formulation with heuristic solver	Yes	No	Affordable	Some VNFs of SFC
Addad et al. [21]	Hybrid environment of core cloud and edge cloud	Users' mobility	save the allocated resources	DDPG and DQN	No	No	Costly due to model training	Not clear
Wei et al. [18]	core cloud	Fluctuations in user resources requirements	Guarantee users' QoE	MDP formulation with Reinforcement Learning based solver	Yes	No	Costly due to model training	Not clear
Zhang et al. [22]	Hybrid environment of core cloud and edge cloud	Unbalanced distribution of remaining physical resources	Release resources of over-loaded nodes	ILP formulation with heuristic solver	Yes	No	Affordable	Only the last VNF of SFC
Proposed Solutions	core cloud	Unbalanced distribution of remaining physical resources	Balance the distribution of remaining physical resources	ILP formulation with heuristic solver	Yes	Yes	Affordable	Total VNFs of SFC

TABLE II
SUMMARY OF NOTATIONS USED.

Notation	Definition
N, E	physical nodes (including forwarding nodes N_F and VNF-enabled nodes N_V) and links respectively
$\tau_{i,j}, B_{i,j}$	transmission delay, bandwidth resources of link (i, j) respectively
C_i	computational resources of a VNF-enabled node i
$\mathrm{s}_r, d_r, \tau_r$	the source node, destination node, and delay requirement of SFC request r, respectively
c_r, b_r	CPU resource requirement for a single VNF, and bandwidth resource requirement of SFC request r
SQ	service queue
$\mathcal{F}_r$	the set of VNFs of SFC request r
π_m^r	the m_{th} VNF of SFC request r
P_r	the transmission path of SFC request r
M_r	the VNFs placement solution of $\mathcal{F}_r$
(r, π_m^r)	the virtual link between virtual node π_m^r and π_{m+1}^r
$h_i(\pi)$	binary variable, indicates whether or not node i is capable of VNF π
$X_{i,r}(\pi_m^r)$	binary variable, indicates whether or not VNF π_m^r is placed onto physical node i
$Z_{i,j}(r, \pi_m^r)$	binary variable, indicates whether or not virtual link (r, π_m^r) is mapped onto physical link (i, j)
β	unit bandwidth consumption of per computational unit
W_{cpu}, W_{bw}	benefit of per bandwidth unit and per computational unit, respectively

where s_r, d_r denote the source and destination of r, $\mathcal{F}_r$ denotes the SFC requirement, and τ_r, c_r, b_r denote the latency requirement, the CPU resource requirement for VNFs, and the bandwidth resource requirement, respectively. For example, in Fig. 1, SFC request 3 can be denoted as $r_3 = (\mathrm{S}_2, \mathrm{S}_6, \{\mathrm{VNF1}, \mathrm{VNF2}, \mathrm{VNF3}\}, \tau_3, c_3, b_3)$.

C. Problem Formulation

Now we formally propose the mathematical formulation of the SFC migration problem. We start with the constraints that MANO should obey when recomputing the resource allocation solutions for SFC requests in the service queue.

We first model the VNFs mapping SFC requests, and define some binary variables. $h_i(\pi)$ indicates whether or not node i is capable of VNF π. $X_{i,r}(\pi_m^r)$ indicates whether or not VNF π_m^r is placed onto physical node i, where π_m^r represents the m_{th} VNF of SFC request r. To ensure VNFs are placed onto the VNF-enabled physical nodes, we have:

$$X_{i,r}(\pi_m^r) \leq h_i(\pi_m^r), \forall i \in N, \forall r \in SQ, \forall \pi_m^r \in \mathcal{F}_r. \quad (1)$$

Referring to some relevant works [20], [22], VNFs are not allowed to be split in our current work, so each VNF of a SFC request can be only placed onto one VNF-enabled physical node. Thus, we have the following constraint:

$$\sum_i X_{i,r}(\pi_m^r) = 1, \forall r \in SQ, \pi_m^r \in \mathcal{F}_r. \quad (2)$$

In addition, we also require that each VNF-enabled physical node provides at most one VNF for each $\mathcal{F}_r$, and this can be ensured if Eq. (3) holds.

$$\sum_{\pi_m^r \in \mathcal{F}_r} X_{i,r}(\pi_m^r) \leq 1, \forall i \in N_V, r \in SQ. \quad (3)$$

Then we consider the virtual link mapping of SFC requests. We do not allow path splitting for the purpose of avoiding coordination overhead. Similar to [20], we denote P_r as the transmission path of request r, and M_r as the VNFs placement solution of $\mathcal{F}_r$. Each transmission path should obey the following constraints.

$$\tau_r \geq \sum_{(i,j)\in P_r} \tau_{(i,j)}, \forall r \in SQ. \tag{4}$$

$$M_r \in P_r, \forall r \in SQ. \tag{5}$$

Constraint (4) is latency related. The total delay of the transmission path should be less than the delay required by the request. Constraint (5) requires that the transmission path must pass through the physical nodes that provide the VNF service for request r.

Finally, we consider the constraints of the resource upper bounds. Similarly, we define a binary variable $Z_{i,j}(r, \pi_m^r)$ to indicate whether (r, π_m^r) is mapped on a physical link (i, j) or not, where (r, π_m^r) represents the virtual link between virtual node π_m^r and π_{m+1}^r. Specially, (r, π_0^r) represents the virtual link from s_r to the first VNF, and $(r, \pi_{L_r}^r)$ represents the virtual link from the last VNF to d_r. As mentioned above, we ignore the resource consumption of the forwarding function, so the total computational resource consumed on a VNF-enabled physical node i and the total bandwidth resource consumed on a physical link (i, j) are:

$$R_C(i) = \sum_r \sum_{\pi_m^r} X_{i,r}(\pi_m^r) c_r \tag{6}$$

and

$$R_B(i,j) = \sum_r \sum_{\pi_m^r} Z_{i,j}(r, \pi_m^r) b_r, \tag{7}$$

respectively, where c_r represents the CPU resource requirement for a single VNF of request r, and b_r represents the bandwidth resource requirement of request r.

Similar to [10], [34], we assume that within a request r, the computational resource requirements of VNFs are the same. Thus, we have the following capacity constraints for VNF-enabled physical nodes and physical links.

$$R_C(i) \leq C_i, \forall i \in N_V. \tag{8}$$

$$R_B(i,j) < B_{i,j}, \forall (i,j) \in E. \tag{9}$$

Now, let us use the standard deviation σ_G to measure the distribution of physical resources. σ_G can be calculated as:

$$\begin{aligned} \sigma_G = & \sqrt{\frac{1}{NUM(N_V)} \sum_{i\in N_v} \left(R_C(i) - \overline{R_C}\right)^2} + \\ & \sqrt{\frac{1}{NUM(E)} \sum_{(i,j)\in E} \left(R_B(i,j) - \overline{R_B}\right)^2}, \end{aligned} \tag{10}$$

where $NUM(N_V)$ and $NUM(E)$ represent the number of VNF-enabled physical nodes and physical links in the substrate network G, respectively. $\overline{R_C}$ and $\overline{R_B}$ represent the average value of the consumed resource in N_V and E, respectively. In Fig. 1, after migration, since the overloaded nodes and links no longer exist in the substrate network, σ_G can be reduced.

During SFC migration, we also have to consider the migration cost. Referring to some related works [18], [20], [22], we introduce the migration cost o_r of a SFC request r to represent the extra cost of continuing processing the total amount of the unprocessed packets of r. We use t_r to indicate the time when the migration of r starts, $t_r^{'}$ to indicate the time when the migration of r ends and $u_r(t)$ to represent the flow rate of r at time t. Thus, during migration, the total amount of the unprocessed packets of r can be calculated as $\int_{t_r}^{t_r^{'}} u_r(t)$. Referring to the definition of migration cost in [22], we also use the bandwidth resource requirement of r to represent $u_r(t)$. Since the flow of r should be orderly-executed based on a set of VNFs, the further up in the SFC the VNFs we intend to migrate are, the larger the effects they will bring to the whole SFC [22]. We use $Len(P_r^o, P_r^m)$ to denote how many new virtual links of unprocessed packets of r should be transmitted after migration. For example, in Fig. 1, the original transmission path of SFC request 2 P_2^o is (S1 → V1 → V2 → V4 → S5), and its transmission path after migration P_2^m is (S1 → V1 → V3 → S4 → V4 → S5). In such case, $Len(P_2^o, P_2^m) = 4$, because after virtual node V1, subsequent links begin to change. Therefore, we define o_r as:

$$o_r = \int_{t_r}^{t_r^{'}} Len(P_r^o, P_r^m) b_r. \tag{11}$$

In addition, to calculate the total migration cost $\mathcal{O}_G$, we need to accumulate the migration cost of each SFC request in the service queue, and the total migration cost can be represented as:

$$\mathcal{O}_G = \sum_{r\in SQ} o_r \tag{12}$$

Finally, taking the distribution of physical resources as well as the migration cost into account, we define the SFC migration problem as:

$$min\ \sigma_G + \mathcal{O}_G \tag{13}$$

$$s.t.\ (1-5), (9-10) \tag{13.1}$$

$$h_i(\pi_m^r) \in \{0,1\}, \forall i \in N, \forall r \in SQ, \pi_m^r \in \mathcal{F}_r \tag{13.2}$$

$$X_{i,r}(\pi_m^r) \in \{0,1\}, \forall i \in N, r \in SQ, \pi_m^r \in \mathcal{F}_r \tag{13.3}$$

$$Z_{i,j}(r, \pi_m^r) \in \{0,1\}, \forall (i,j) \in E, r \in SQ, \pi_m^r \in \mathcal{F}_r \tag{13.4}$$

IV. Heuristics for SFC migration

In this section, we elaborate on the conservative migration strategy inspired by [20], [22] and our proposed aggressive migration strategy, respectively.

A. A conservative migration strategy

To quickly restore the user's service or reduce the effect caused by migration on the whole SFC, some researchers [20], [22] are prone to migrate only the last VNF of the SFC. Inspired by this idea, we designed a conservative migration strategy that is performed through the following steps.

Step 1: fetch request with low resource requirement. The smaller the resource requirement of a request, the more candidate resource allocation solutions there will be. We calculate the resource requirement of a request through Eq. (14) (In this paper, we prioritize computational resources due to their higher profit). Before migration, we sort all requests in SQ by their $\mathcal{Y}_r$ and select the request with the smallest $\mathcal{Y}_r$ for migration.

$$\mathcal{Y}_r = \sum_{\pi \in r} c_r \tag{14}$$

Step 2: resource recovery. Before SFC migration, MANO should recover the physical resources that have been allocated to request r, including computational resources in the VNF-enabled physical node and bandwidth resources in the physical link.

Step 3: determine the stable section. Since the conservative migration strategy only migrates the last VNF of the SFC, MANO needs to determine the stable section of the original resource allocation solution. We use m_r^s, p_r^s to represent the stable section of M_r^o and the stable section of P_r^o, respectively. Then we denote L_r as the length of $\mathcal{F}_r$, $m_{L_r}^r$ as the last element of M_r^o, and denote m_{pen}^r as the penultimate element of M_r^o. Clearly, m_r^s equals to $M_r - m_{L_r}^r$, and p_r^s is $(s_r \to ... \to m_{pen}^r)$.

Step 4: recalculate the resource solution for the migration section. This step needs to solve the following two problems. (a) Where the last VNF of $\mathcal{F}_r$ should be migrated to. (b) Which transmission path from m_{pen}^r to d_r should be chosen. We use $\pi_{L_r}^r$ to represent the last VNF of $\mathcal{F}_r$, m_r^{re}, p_r^{re} to represent the new resource allocation solution for $\pi_{L_r}^r$, and the new allocation solution for the transmission path from m_{pen}^r to d_r, respectively. To avoid loops, we first need to remove elements of p_r^s from G (except for m_{pen}^r for the purpose of path calculation [1]) and get the pruned substrate network G_{p_1}. Then, with the constraints of (1), (3), we calculate the candidate physical node set $\mathcal{N}_r$ for $\pi_{L_r}^r$. Typically, $\mathcal{N}_r = N_V^{'} - (N_V^{'} \cap m_r^s)$, where $N_V^{'}$ represent the set of VNF-enabled physical nodes in G_{p_1}. Then, we select the physical node i_{most} that not only can provide service for $\pi_{L_r}^r$, but also with the most remaining computational resource in $\mathcal{N}_r$, and place $\pi_{L_r}^r$ on i_{most}. Next, we remove the physical links that cannot meet the bandwidth requirement of r in G_{p_1}, and get another pruned substrate network G_{p_2}. We can get all paths from m_{pen} to d_r in G_{p_2} with a modified Depth First Search (DFS) method [35]. Then we select the paths that obey constraint (4), (5) to form the set $\mathcal{P}_r$ of feasible solutions to p_r^{re}, and place the virtual path from m_{pen}^r to d_r on the path p_{least} with the least standard deviation of the remaining bandwidth resources in $\mathcal{P}_r$. After the above process, we get m_r^{re} equal to i_{most} and p_r^{re} equal to p_{least}.

Step 5: resource remapping. Once MANO gets p_r^s, p_r^{re}, m_r^s, m_r^{re}, it can determine a new resources allocation solution for request r. Then, MANO should remap the resource for request r based on P_r^m, M_r^m.

Algorithm 1 shows the pseudo-code of the conservative migration strategy. The Resources Recovery (RR) procedure is used to carry out step 2, and the Recalculate Resource Solution (RRS) procedure is used to carry out step 4. Step 5 is carried out through line 8 to line 13.

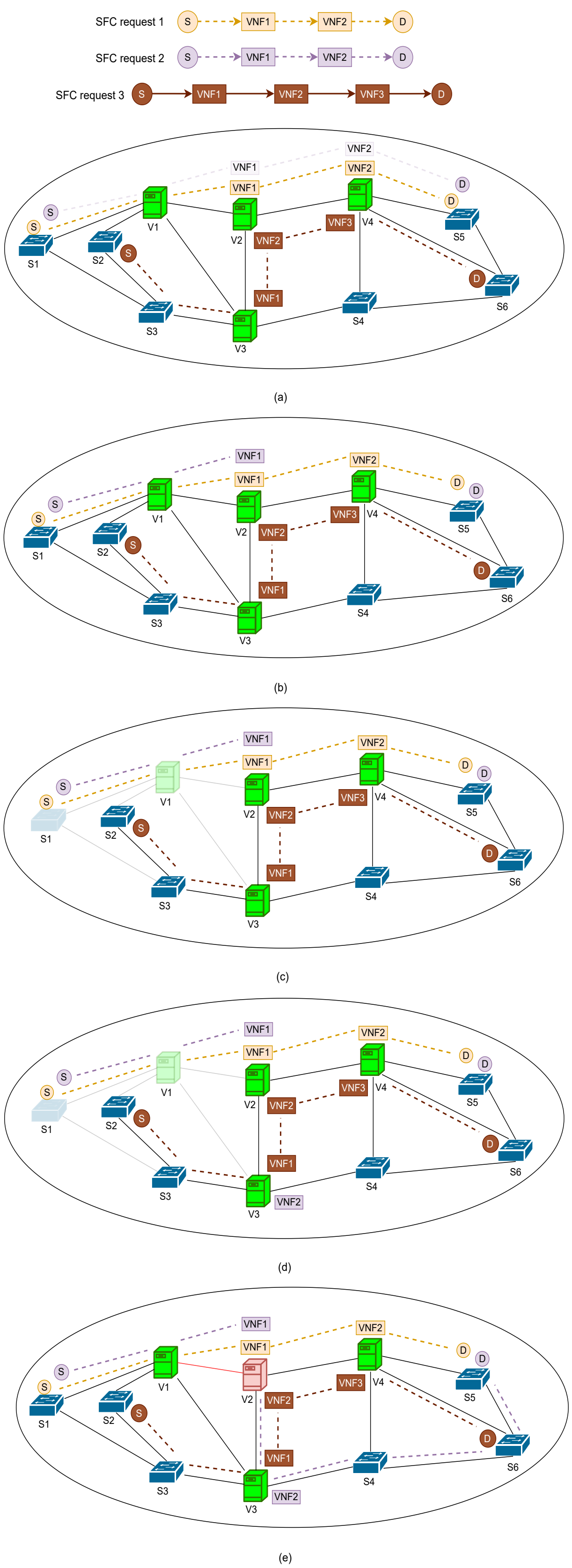
In Fig. 1, since the resource requirements of SFC request 1 and 2 are relatively small, MANO will migrate them first. Here, we use the migration process of SFC request 2 to illustrate the conservative migration strategy. Firstly, MANO needs to recover the physical resources previously allocated to SFC request 2, as shown in Fig. 2(a). Secondly, MANO needs to determine the stable section of SFC request 2's original resource allocation solution, as shown in Fig. 2(b), where m_r^s is (S1 → V1 → V2), and p_r^s is {V2}. Thirdly, MANO has to determine where the last VNF (VNF2) of $\mathcal{F}_2$ should be migrated to, and which transmission path from m_{pen}^2 to d_2 should be chosen. To solve the above problems, MANO first removes some nodes and links in the substrate network to avoid loops, as shown in Fig. 2(c). Then, since V3 has the most remaining computing resources, MANO maps VNF2 on V3 when constraint (1) is met and sets m_2^{re} to V3, as shown in Fig. 2(d). After that, MANO calculates feasible paths from m_{pen}^2 to d_2 in the pruned substrate network. Under constraints (4), (5), since the standard deviation of the bandwidth resource distribution of path (V2 → V3 → S4 → S6 → S5) is the smallest, MANO maps p_2^{re} on it. Finally, MANO reallocates the resource for SFC request 2 based on $m_2^s \cup m_2^{re}$, $p_2^s \cup p_2^{re}$, and the migration effect is shown in Fig. 2(e).

Next, we analyze the complexity of the conservative migration strategy. First, sorting all requests (line 2) requires $O(|SQ| \log_2 |SQ|)$ computation. Then, the first **while**-loop (line 3) terminates in $|SQ|$ iterations. For the RR procedure, it needs $O(|E|+|N_V|)$ computation. For the RRS procedure, its first **for**-loop (line 2) terminates in $|p_r^s| - 1$ iterations, and its second **for**-loop (line 8) terminates in $|P_r^c|$ iterations. In addition, for the modified DFS method, a simple path can be found in $O(|E| + |N|)$ [35], which thus causes $O(|P|\,(|E| + |N|)$ computation to calculate all the paths from node s to node d in graph G, where P, E, N denote the set of candidate paths from s to d, the set of physical links in G, and the set of physical nodes in G, respectively. In our work, we let E_r, N_r to represent the physical links set and physical nodes sets in each G_{p2}, and we let $E_{con} = \max_{r=1} |E_r|$, $N_{con} = \max_{r=1} |N_r|$, $A_{con} = \max_{r=1}(|p_r^s| - 1)$, $B_{con} = \max_{r=1} |P_r^c|$. Thus, the computation of RRS procedure is $O(A_{con} + B_{con}(E_{con} + N_{con}) + B_{con})$. The last two **for**-loops (line 8 and line 11) in the conservative migration strategy need to terminate in $|M_r^m|$, $|P_r^m|$ iterations, respectively. Since $B_{con}(E_{con} + N_{con})$ is greater than $\log_2 |SQ|$, A_{con}, $(|E|+|N|)$, $|M_r^m|$, and $|P_r^m|$, the overall time complexity of conservative migration strategy is $O(|SQ| \log_2 |SQ| + |SQ|\,((|E|+|N_V|) + A_{con} + B_{con}(E_{con} + N_{con}) + B_{con} + |M_r^m| + |P_r^m|)) = O(|SQ|\, B_{con}(E_{con} + N_{con}))$.

B. An aggressive migration strategy

In order to reduce the migration cost, the conservative migration strategy only migrates the last VNF of the SFC, and this will cause overloaded nodes and links in the substrate

[1] In the following step, MANO should calculate the paths from m_{pen}^r to d_r in G_{p_1}, thus, m_{pen}^r should be included in G_{p_1}.



(a)

(b)

(c)

(d)

(e)

Fig. 2. Conservative migration process of a single SFC request

Algorithm 1 Conservative Migration Algorithm

Input: initial substrate network G, information of the requests in service queue SQ, including $c_r, b_r, \mathcal{F}_r, P_r^o, M_r^o, s_r, d_r, \tau_r$.
Output: substrate network G' after SFC migration, and migration cost $\mathcal{O}_G$.

```
Initialization: let $\mathcal{O}_G = 0$;
Sort all $r \in SQ$ by its $\mathcal{Y}_r$ in ascending order;
while $SQ \neq \emptyset$ do
    Select $r$ with the minimal $\mathcal{Y}_r$, call RR procedure, then let $G = G_{rec}$, $M_r^m = \{\}$, $P_r^m = \{\}$;
    Let $m_r^s = M_r - m_{L_r}^r, p_r^s = (s_r, ..., m_{pen}^r)$;
    Call RRS procedure, then get $p_r^{re}, m_r^{re}$;
    Let $M_r^m = m_r^s \cup m_r^{re}$, $P_r^m = p_r^s \cup p_r^{re}$;
    for each node $i \in M_r^m$ do
        $C_i = C_i - c_r$;
    end for
    for each link $(i,j) \in P_r^m$ do
        $B_{i,j} = B_{i,j} - b_r$;
    end for
    Calculate migration cost $o_r$ of request $r$ according to Eq.(11);
    $\mathcal{O}_G = \mathcal{O}_G + o_r$, $SQ = SQ - r$, and then migrate the next SFC request;
end while
return $G', \mathcal{O}_G$.
```

Procedure 1 Resources Recovery (RR)

Input: $G, b_r, c_r, P_r^o, M_r^o$.
Output: G_{rec} (substrate network after resources recovery).

```
for each link $(i,j) \in E$ do
    if $(i,j) \in P_r^o$ then
        $B_{i,j} = B_{i,j} + b_r$;
    end if
end for
for each node $i \in N_V$ do
    if $i \in M_r^o$ then
        $C_i = C_i + c_r$;
    end if
end for
return $G_{rec}$.
```

network to still exist after SFC migration, as shown in Fig. 2(e). To distribute physical resources more evenly, we propose an aggressive migration strategy that migrates the total SFC, which is performed through the following steps.

Step 1, Step 2 are the same as *Step 1 and Step 2* of conservative migration strategy.

Step 3:calculate a new resources allocation solution. To solve the SFC migration problem, we split it into VNF migration and virtual link migration. For VNF migration, we greedily place VNFs on the physical nodes with the most remaining resources. For virtual link migration, we first remove the physical links that cannot meet the bandwidth requirement of r and get a pruned network G_{p_1}, and then

Procedure 2 Recalculate Resource Solution (RRS)

```
Input: $G, b_r, c_r, p_r^s, m_r^s, d_r$.
Output: $p_{re}^m$, $m_r^{re}$.
Initialization: let $s_r^{'}$ equal the last element of $p_r^s$, candidate transmission path $\mathcal{P}_r$={}, $\mathcal{N}_r = N_V^{'} - (N_V^{'} \cap m_r^s)$;
for each node $i \in (p_r^s - s_r^{'})$ do
    Remove $i$ and its directly connected links form $G$;
end for
Remove the physical links that can not meet the bandwidth requirement of $r$.
Select the node $i_{most}$ with the most remaining resources in $\mathcal{N}_r$, let $m_r^{re} = i_{most}$;
Run modified Depth First Search with $s_r^{'}, d_r, G$, then generate a set $P_r^c$ of candidate paths;
for each path $p \in P_r^c$ do
    if $(p \cup p_r^s)$ allows constraints (4),(5) then
        $\mathcal{P}_r = \mathcal{P}_r + (p \cup p_r^s)$
    end if
end for
Select the path $p$ with the smallest standard deviation of the remaining bandwidth resources from $\mathcal{P}_r$, then let $p_r^m = p$;
return $p_r^{re}$, $m_r^{re}$.
```

we get all paths from s_r to d_r in G_{p_1} through a modified DFS method and select the paths that obey constraints (4), (5) to form the set $\mathcal{P}_r$ of feasible solutions to P_r^m. Finally, we place P_r^m on the path p with the smallest standard deviation of the remaining bandwidth resources in $\mathcal{P}_r$.

Step 4: resources remapping. MANO needs to remap the resources based on the new resources allocation solution calculated by *step 3*.

The procedure of aggressive migration is shown in Alg. 2. Line 5 to line 16 are used to calculate new resources allocation solutions P_r^m, M_r^m for request r. And step 4 is carried out through line 17 to line 22. Similarly, we take the migration process of SFC request 2 in Fig. 1 as an example and the process shown in Fig. 3. Fig. 3(a) corresponds with the process of resource recovery, Fig. 3(b) corresponds with the process of VNF migration, and Fig. 3(c) corresponds with the process of virtual link migration and resource remapping.

Similarly, we take the migration process of SFC request 2 in Fig. 1 as an example. As shown in Fig. 3(a), MANO should also recover the physical computational and bandwidth resources previously allocated to SFC request 2. Then, MANO greedily places the VNFs of $\mathcal{F}_2$ on the physical nodes with the most remaining resources, and sets M_2^m to {V1, V2}, as shown in Fig. 3(b). After that, MANO first removes the physical links that cannot meet the bandwidth requirement of SFC request 2 in the substrate network, and then it calculates the feasible paths from s_2 to d_2 in the pruned substrate network under constraints (4), (5). Since the standard deviation of the bandwidth resource distribution of path (S1 → V1 → V3 → S4 → S6 → S5) is the smallest, MANO sets P_2^m to it. Finally, MANO reallocates the resource for SFC request 2 based on M_2^m, P_2^m, and the migration effect is shown in Fig. 3(c). Note that, after aggressive migration, overloaded nodes and links no longer exist in the substrate network, so the aggressive migration strategy is expected to make the distribution of physical resources more balanced.

Likewise, we analyze the time complexity of our proposed aggressive migration strategy. Similarly, the computation required for sorting all requests (line 2) is $O(|SQ| \log_2 |SQ|)$, the first **while**-loop (line 3) terminates in $|SQ|$ iterations, and the RR procedure needs $O(|E| + |N_V|)$ computation. The first (line 5), second (line 11), third (line 17) and fourth (line 20) **for**-loops need to terminate in $|\mathcal{F}_r|$, $|P_r^c|$, $|P_r^m|$, and $|M_r^c|$ iterations, respectively. Then, we also let E_r, N_r to represent the physical links set and physical nodes set in each G_{p1}, and let $\mathcal{F} = \max_{r=1} |\mathcal{F}_r|$, $E_{agg} = \max_{r=1} |E_r|$, $N_{agg} = \max_{r=1} |N_r|$, $B_{agg} = \max_{r=1} |P_r^c|$. Thus, the time complexity of the aggressive migration strategy is $O(|SQ| \log_2 |SQ| + |SQ| (\mathcal{F} + B_{agg}(E_{agg} + N_{agg}) + B_{agg} + |P_r^m| + |M_r^m|)) = O(|SQ| B_{agg}(E_{agg} + N_{agg}))$.

Note that, before running the modified DFS method, the conservative migration strategy removes more nodes and links from the substrate network, so its B_{con}, E_{con}, and N_{con} are smaller than those of the aggressive migration strategy. Therefore, the time complexity of the conservative migration strategy is slightly smaller than that of the aggressive migration strategy.

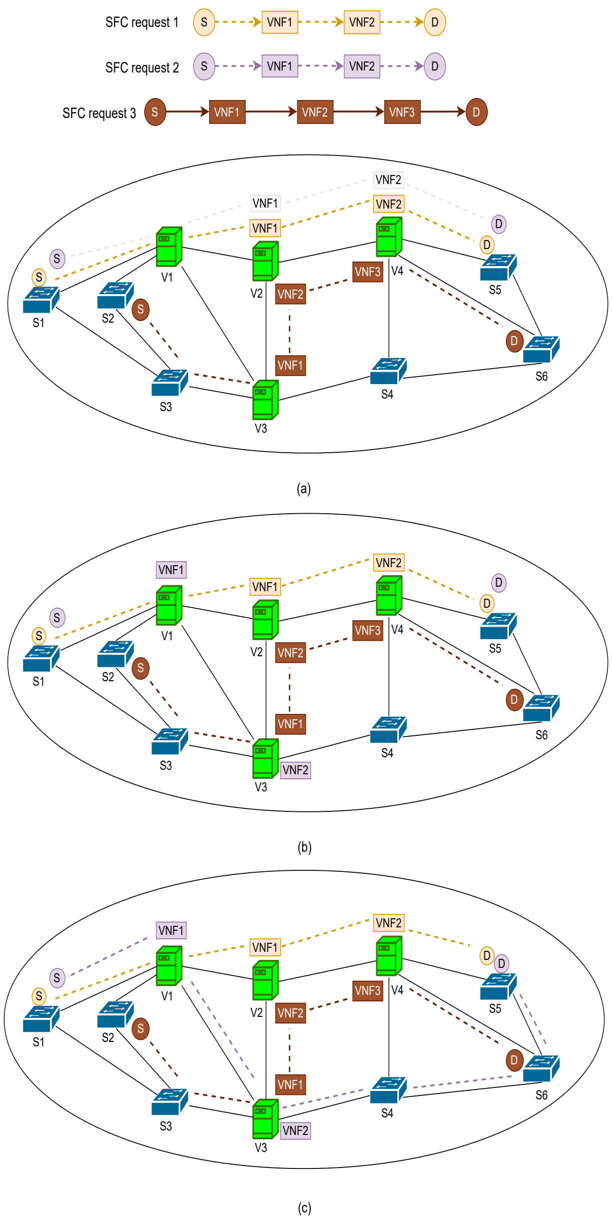
V. Numerical results

In this section, we demonstrate the performance evaluation of our proposed migration strategy. We first discuss the simulation setup used to evaluate the algorithms in our work. We then introduce the state-of-the-art heuristics for allocating resources for subsequent SFC requests. Finally, we compare our aggressive migration strategy with the conservative migration strategy and describe our main simulation results and analysis.

A. Simulation settings

In our simulations, we construct a substrate network as is shown in Fig. 4, which is widely used in network slicing and SFC research, such as [17], [18]. The parameters of the substrate network are listed in Table III. In the substrate network, there are 7 VNF-enabled physical nodes, and they can provide all kinds of VNF services. Currently, we do not consider the difference in the resource upper limit between different VNF-enabled physical nodes and physical links. Therefore, the computational resources of VNF-enabled physical nodes are set to 10000 units. As some researchers assume that processing one unit of flow requires one unit of computational capacity [38], we set the bandwidth resources of physical links to 10000 units too. In addition, the latency of physical links was set to 1ms.

The parameters of various SFC requests in our work are shown in Table III. In real-world scenarios, there will be permanent and temporary network service requests [36], [37]. Therefore, we designed a type of permanent request, which



(a)

(b)

(c)

Fig. 3. Process of aggressive migration scheme

TABLE III
NETWORK PARAMETERS

Parameters	Value
Number of nodes	15
Number of links	27
Number of VNFs	7
β	1
Capacity of VNF enabled nodes	1×10^4
Capacity of links	1×10^4
Latency of links	1

Algorithm 2 Aggressive Migration Algorithm

Input: initial substrate network G, information of the requests in service queue SQ, including $c_r, b_r, \mathcal{F}_r, P_r^o, M_r^o, s_r, d_r, \tau_r$.
Output: substrate network G' after SFC migration, and migration cost $\mathcal{O}_G$.

```
Initialization: let $\mathcal{O}_G = 0$;
Sort all $r \in SQ$ by its $\mathcal{Y}_r$ in ascending order;
**while** $SQ \neq \emptyset$ **do**
    Select $r$ with the minimal $\mathcal{Y}_r$, call RR procedure, then let $G = G_{rec}$, $M_r^m = \{\}$, candidate VNF-EN node set $\mathcal{N}_r = N_V$, candidate transmission path $\mathcal{P}_r = \{\}$;
    **for** each VNF $f_i$ in $\mathcal{F}_r$ **do**
        Select the node $i_{most}$ with the most remaining resources from $\mathcal{N}$;
        $M_r^m = M_r^m + i_{most}$, $\mathcal{N}_r = \mathcal{N}_r - i$;
    **end for**
    Copy $G$, get duplicate $G_{p_1}$ and remove the physical links that can not meet the bandwidth requirement of request $r$ in $G_{p_1}$;
    Run modified Depth First Search with $s_r, d_r, G_{p_1}$, then generate a set of candidate paths $P_r^c$;
    **for** each path $p \in P_r^c$ **do**
        **if** $p$ allows constraints (4), (5) **then**
            $\mathcal{P}_r = \mathcal{P}_r + p$;
        **end if**
    **end for**
    Select the path $p$ with the smallest standard deviation of the remaining bandwidth resources from $\mathcal{P}_r$, then let $P_r^m = p$;
    **for** each link $(i, j) \in P_r^m$ **do**
        $B_{i,j} = B_{i,j} - b_r$
    **end for**
    **for** each node $i \in M_r^m$ **do**
        $C_i = C_i - c_r$
    **end for**
    Calculate migration cost $o_r$ of request $r$ according to Eq.(11);
    $\mathcal{O}_G = \mathcal{O}_G + o_r$, $SQ = SQ - r$, and then migrate the next SFC request;
**end while**
**return** $G', \mathcal{O}_G$.
```

will occupy a large amount of physical resources, but correspondingly, their number was set to be scarce. The other two types of requests are temporary requests. They take up fewer physical resources than permanent requests, but their number is very large. What is more, the bandwidth resource requirement of request r is related to its computational resource requirement, so we get $b_r = \beta \cdot c_r$. As we also assume that processing one unit of data flow requires one unit of computational capacity [38], we set $\beta = 1$ in our current work. The source and destination of all SFC requests are randomly generated from the S and D nodes in the substrate network.

We assume that at the initial moment, there are 6 permanent

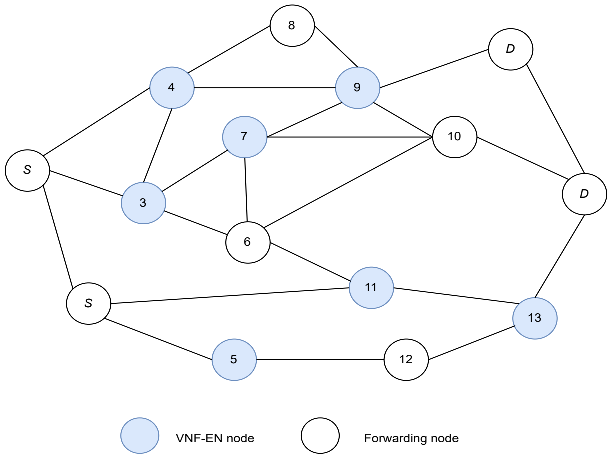


Fig. 4. Substrate network for simulation

TABLE IV
PARAMETERS OF SFC REQUESTS

Parameters	Type1	Type2	Type3
Number of VNFs	{2,3,4,5,6}	{1,2,3}	{2,3,4,5}
CPU resources requirement c_r for VNF (unit)	U(900,1000)	U(10,20)	U(100,200)
Bandwidth resources requirement b_r (unit)	$\beta \cdot c_r$	$\beta \cdot c_r$	$\beta \cdot c_r$
Latency requirement (ms)	$\leqslant$20	$\leqslant$30	$\leqslant$ 40
Average lifetime (time slot)	permanent	$\geqslant$100	$\geqslant$100

requests in the service queue. Under the premise of satisfying resource mapping constraints, the resource allocation solutions of these requests are random, so as to simulate the randomness of the initial physical resources distribution. After the initial moment, we assume that the subsequent requests consist of Type 2 and Type 3 requests randomly, and these SFC requests are issued following a Poisson process of rate λ, where λ is set to 5 in our work.

The migration strategies are evaluated using the Python programming language and an extended package for graph theory called networkx [39].

B. Heuristics for subsequent SFC requests

The BestFit [23], [24] and CN [25] heuristics are employed for allocating resources for the subsequent SFC requests, and below we give a brief description of them.

- BestFit: First, the physical node with the most remaining computational resources is greedily selected to place the VNFs requested by a request, and then the shortest path between these nodes is calculated to connect these VNFs in series under the premise of meeting the bandwidth requirements.
- CN: CN similarly divides resource allocating into two stages. First, it calculates the importance of each node according to the degree, betweenness centrality of the nodes, the remaining computational resources of the nodes, and the remaining bandwidth resources of the links to which the nodes are directly connected. Then, the VNFs requested by SFC requests are greedily placed on physical nodes with high importance. Finally, under the premise of meeting the bandwidth requirements, the shortest path between these physical nodes is calculated to connect the VNFs in series.

C. Evaluation indicators

- Standard deviation of physical resources distribution: The standard deviation of physical resource distribution is used to measure the balance of physical resource distribution, which can be calculated by Eq. (10).
- Total migration cost: The total migration cost is used to measure the total cost of migrating the SFC requests in the service queue, which can be calculated by Eq. (12).
- Acceptance ratio: The SFC request acceptance ratio at time slot T can be defined as:

$$AC(T) = \frac{\sum_{t=0}^{T} NUM\_NSR\_S}{\sum_{t=0}^{T} NUM\_NSR}, \tag{15}$$

where NUM_NSR_S and NUM_NSR are the number of SFC requests that are successfully allocated resources and the number of total SFC requests.

- Node utilization: The utilization of the computational resource of VNF-enabled physical nodes at time slot T can be defined as:

$$NU(T) = \frac{\sum\limits_{r\in SQ_T} \sum\limits_{\pi\in\mathcal{F}_r} c_r}{\sum\limits_{i\in N_V} C_i}. \tag{16}$$

- Link utilization: The utilization of the bandwidth resource of physical links at time slot T can be defined as:

$$LU(T) = \frac{\sum\limits_{r\in SQ_T} \sum\limits_{(i,j)\in P_r} b_r}{\sum\limits_{(i,j)\in E} B_{i,j}} \tag{17}$$

- Long-term profit: The total profit earned by the network operator for serving SFC requests from the initial time to time slot T can be defined as:

$$Pro(T) = \int_{t=0}^{T} \sum_{r\in SQ_t} (W_{cpu} \sum_{\pi\in\mathcal{F}_r} c_r + W_{bw} \sum_{(r,\pi_m^r)\in P_r^V} b_r), \tag{18}$$

where W_{cpu} and W_{bw} represent the benefit of per computational unit and per bandwidth unit, respectively, and P_r^V denotes the virtual path of request r. In our work, W_{cpu}, and W_{bw} are set to 0.1, and 0.05, respectively.

D. Simulation results and analysis

We first compare the changes in resource distribution in the substrate network after two different SFC migration strategies. As shown in Fig. 5, in a single experimental instance, our proposed migration strategy can make the initial physical resource distribution more balanced. We then randomly generate 20 other instances of the problem (13). Fig. 6 demonstrates the effectiveness of our proposed strategy and compared with the conservative migration strategy, the σ_G of our proposed strategy is 14.991% lower on average.

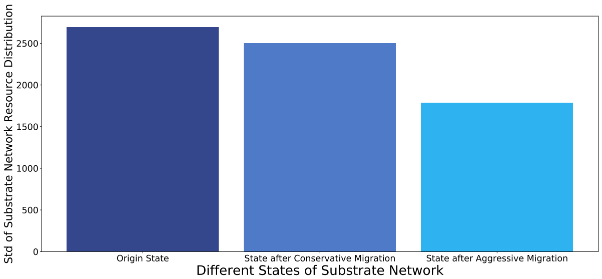


Fig. 5. Different standard deviations of an experimental instance

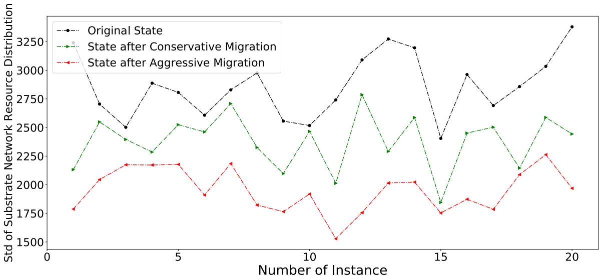


Fig. 6. Std of 20 experimental instances

However, as our proposed strategy migrates the total SFC, it causes more migration cost, as shown in Fig. 7. In 20 experimental instances, the migration cost of our proposed strategy is on average 25.5% higher than with the conservative migration, as shown in Fig. 8. The above results are summarized in Table V.

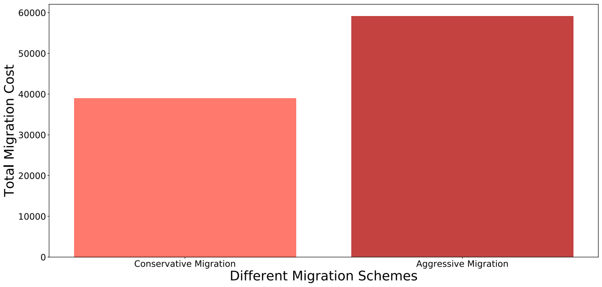


Fig. 7. Different migration cost for an experimental instance

TABLE V
AVERAGE STD AND MIGRATION COST OF 20 EXPERIMENTAL INSTANCES

Substrate network state	**Average std of substrate network resources**	**Average migration cost**
Initial	2863.012	/
After conservative migration	2379.958	44176.363
After aggressive migration	1950.773	55441.179

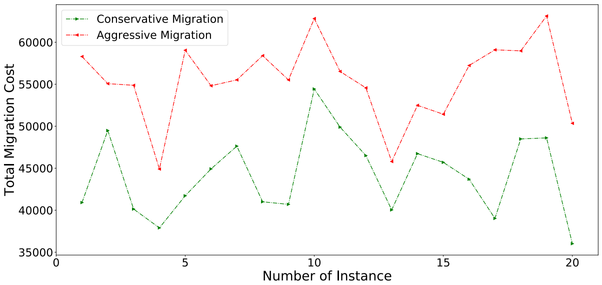


Fig. 8. Migration cost for 20 experimental instances

Next, we compare the changes in the acceptance ratio of subsequent SFC requests, the utilization of physical resources, and the long-term total profit, before and after SFC migration.

We first analyze the simulation results for a single experimental instance. As shown in Fig. 9, and 10, it can be seen that as time goes by, MANO receives more and more SFC requests, and the physical resources become more and more strained. Therefore, after a period of time, the acceptance ratio of SFC requests starts to decrease. By migrating SFC requests in the service queue at the initial moment through our proposed strategy, we can make the operators receive more SFC requests, and thus increase the final acceptance ratio. However, the conservative migration strategy has a negative impact on the final acceptance ratio. We then analyze changes in physical resource utilization. As shown in Fig. 11, 12, and 13, for both “BestFit” and “CN” heuristics, our proposed migration strategy improves final resource utilization and long-term profit. The “BestFit” heuristic, although the conservative migration strategy reduces the acceptance ratio of SFC requests, improves the final resource utilization. For the “CN” heuristic, the conservative migration strategy reduces both the acceptance ratio of SFC requests and the final resource utilization. Interestingly, however, the conservative migration strategy advances the time when the physical resource utilization reaches a plateau. From time slot 30 to time slot 44, the resource utilization of the “Conservative Migration + CN” strategy is higher than the pure “CN” strategy. Although the resource utilization of the pure “CN” strategy exceeds that of the “Conservative Migration + CN” strategy from time slot 45, it still needs a long period of time for the pure “CN” strategy to catch up with the long-term profit of the “Conservative Migration + CN” strategy. From time slot 45 to time slot 60, the gap between the pure “CN” strategy and the “Conservative Migration + CN” strategy in long-term profit gradually narrows, but until time slot 60, the long-term profit of the “Conservative Migration” strategy is still higher than that of the pure “CN” strategy.

According to Eq. (18), the long-term total profit defined in our work is an obvious time-dependent monotonically increasing function (unless there are no SFC services in the service queue for a certain amount of time, which can be ruled out in our work as there are some permanent SFC requests). Therefore, if MANO no longer receives new SFC requests and no SFC requests leave the service queue, the long-term profits of different schemes will show perfect linear growth with the time slots. In in Fig. 13, the slopes of some straight lines change over time. For example, from time slot 40 to time slot 46, the long-term profit of "Conservative Migration + CN" is actually higher than the long-term profit of "Conservative Migration + BestFit". But after time slot 37, as shown in Fig. 11 and Fig. 12, both the node resources consumption and link resources consumption of "Conservative Migration + BestFit" are higher than that of "Conservative Migration + CN", which results in the long-term profit of "Conservative Migration + BestFit" surpassing the long-term profit of "Conservative Migration + CN" after time slot 49.

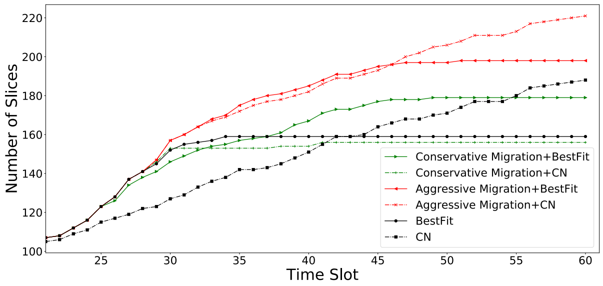


Fig. 9. Number of SFC requests in the service queue over time

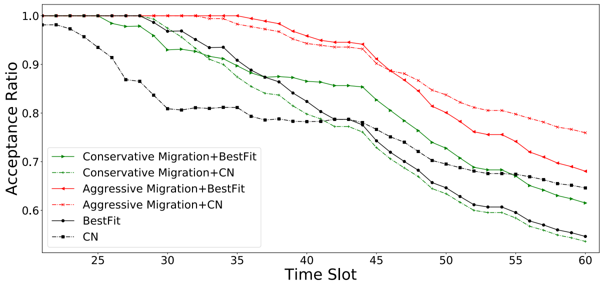


Fig. 10. Different acceptance ratios for an experimental instance

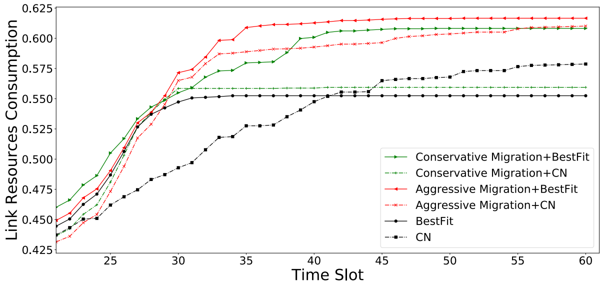


Fig. 11. Different link resources utilization for an experimental instance

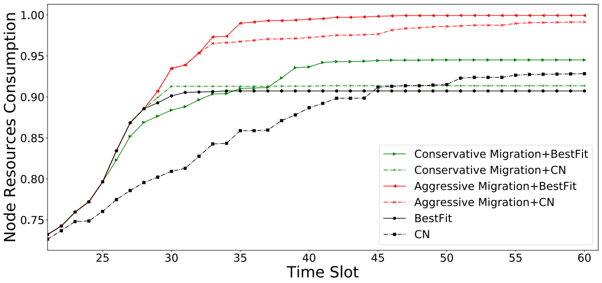


Fig. 12. Different node resources utilization for an experimental instance

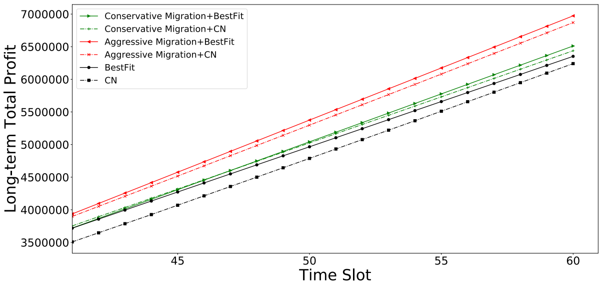


Fig. 13. Different long-term profit for an experimental instance

We then analyze the simulation results for the other 20 experimental instances at time slot 60. As shown in Fig. 14, our proposed aggressive migration strategy improves the acceptance ratio of subsequent SFC requests in most cases (except for instances 4, 5, and 7). Sometimes, MANO may receive a Type 3 SFC request to earn more profit, and in this case it would reject multiple Type 2 SFC requests. Therefore, it is not weird that in a few cases, the acceptance ratio drops after migration. For "BestFit", the average improvement is 11.75%, and for "CN", the average improvement is 6.84%. As shown in Fig. 16, and 17, for all experimental instances, the aggressive migration strategy improves node resource utilization and long-term profit. For "BestFit", node utilization and long-term profit are increased by an average of 8.14% and 9.49%, respectively. For "CN", node utilization and long-term profit are increased by an average of 4.65% and 7.14%, respectively. In some cases (e.g., instance 3), the aggressive migration strategy reduces link resource utilization, but this does not negatively impact long-term profit because no matter how many physical links a virtual link is mapped to, only one virtual link is charged.

Likewise, for all experimental instances, the conservative migration strategy can also improve the operator's long-term profit, but the improvement is smaller than that of aggressive migration strategies. For "BestFit", the average improvement is 3.05%, and for "CN", the average improvement is 4.62%. Similar to the aggressive migration strategy, sometimes the conservative migration strategy reduces the acceptance ratio of SFC requests and physical resource utilization, but still improves the final long-term profit. Overall, with "BestFit", the conservative migration strategy improves the acceptance ratio of SFC requests, node resource utilization, and link resource utilization by 1.82%, 1.59%, and 1.11% on average, respectively. With "CN", the conservative migration strategy improves the acceptance ratio of SFC requests, node resource utilization, and link resource utilization by 2.54%, 1.93%,

TABLE VI
AVERAGE RESULTS AND IMPROVEMENT OF 20 EXPERIMENTAL INSTANCES

Substrate network state & Allocation method	**Average acceptance ratio (improvement)**	**Average node resources utilization (improvement)**	**Average link resources utilization (improvement)**	**Average long-term profit (improvement)**
Initial & BestFit (baseline)	0.5531	0.9117	0.5827	6303710.1509
Initial & CN (baseline)	0.6165	0.9243	0.5729	6268964.1997
After conservative migration & BestFit	0.5713 (1.82%)	0.9276 (1.59%)	0.5938 (1.11%)	6496126.1533 (3.05%)
After conservative migration & CN	0.6419 (2.54%)	0.9436 (1.93%)	0.5903 (1.74%)	6558302.0120 (4.62%)
After aggressive migration & BestFit	0.6706 (**11.75%**)	0.9931 (**8.14%**)	**0.6258 (4.31%)**	6901688.4058 (**9.49%**)
After aggressive migration & CN	0.6849 (6.84%)	0.9708 (4.65%)	0.6012 (2.83%)	6716408.8979 (7.14%)

and 1.74% on average, respectively. The above results are summarized in Table VI.

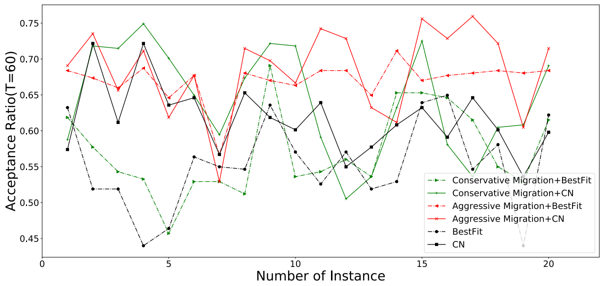


Fig. 14. Acceptance ratio for 20 experimental instances

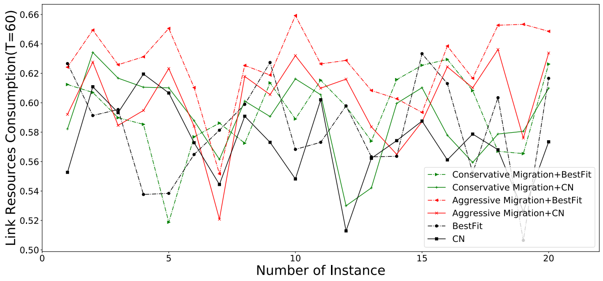


Fig. 15. Link resources utilization for 20 experimental instances

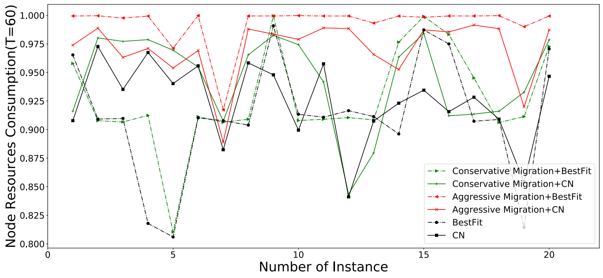


Fig. 16. Node resources utilization for 20 experimental instances

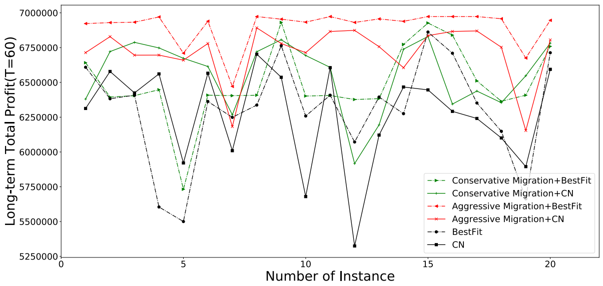


Fig. 17. Long-term total profit for 20 experimental instances

Compared with the conservative migration strategy, the improvements in the evaluation indicators of our proposed migration strategy are better. For "BestFit", the improvements in average acceptance ratio for subsequent SFC requests, average node resources utilization, average link resources utilization, and average long-term profit are 9.93%, 6.55%, 3.2%, 6.44% better than those for the conservative migration strategy. For "CN", these numbers are 4.3%, 2.72%, 1.09%, and 2.52%, respectively. In addition, as shown in Fig. 16, in most cases, node resources utilization rates are close to 100% (except for instance 5, 7) after aggressive migration. Though the migration cost of our proposed strategy is on average 25.5% higher, the migration cost is much smaller than the final long-term profit. For "CN", the average migration costs caused by the conservative migration strategy and the aggressive migration strategy are only 0.7074% and 0.8844% of the original final long-term profit. For "BestFit", those numbers are 0.7008% and 0.8795%, respectively. These migration costs are almost negligible compared to the improvement in the final long-term profit.

VI. CONCLUSION AND FUTURE WORK

This paper investigated the impact of the uneven distribution of substrate network resources on network operators and subsequent SFC requests. We first modeled the SFC migration problem as an ILP, then, inspired by some outstanding relevant works [20], [22], we designed a conservative migration strategy and further proposed an aggressive migration strategy to reduce the imbalance of network resources distribution. After migration, we employed two state-of-the-art heuristics to allocate resources for the subsequent SFC requests. The simulation results not only demonstrated that the unbalanced distribution of physical resources indeed affects both the subsequent SFC requests and the network operators negatively, but also showed the efficiency of our proposed strategy.

Yet, we still have to conduct further research to explore the migration problem. Here, we list some aspects we need to improve in the future.

A. More complex application scenarios

At present, we investigated the SFC migration problem in the core cloud. But for some more complex application scenarios, like network slices requests [40], both the edge cloud servers and the core cloud servers should be considered, as they can form complex graphs of VNFs with different service chains. In such cases, the model we defined in current work is not suitable to meet realistic situations. Therefore, we should rethink some of the constraints and design a more complicated model.

B. Further exploration of the optimization goal

As shown in Fig. 17, although in most cases eliminating the unbalanced physical resources distribution does bring positive effect on the network operators, in a small number of instances (e.g., instance 7), the improvement is not obvious. In addition, our current work does not consider the difference in the resource upper limit between different physical nodes, physical links and the potential effects that the topology structure may cause to our optimization goal. Therefore, to study the specific relationship between the unbalanced distribution of physical resources and the long-term profit of network operators, extensive experiments should be carried out in the future. At that time, we will also consider some interesting attributes of the physical nodes and links, such as the degree of a node and the betweenness centrality, to further adjust our optimization goal.

C. Migration research with prediction mechanisms

In this paper, we do not fully take the volatility of services' resource requirements into account. To further guarantee services' QoE or utilize the physical resources, an effective prediction mechanism is required. However, predicting various kinds of services' resource requirements is particularly challenging, because their traffic characteristics are different. To better carry out migration research, we will explore how to design a general artificial intelligence model to predict the resource requirements of various differentiated services in the future.

References

[1] Taleb T, Kunz A. Machine type communications in 3GPP networks: potential, challenges, and solutions[J]. IEEE Communications Magazine, 2012, 50(3): 178-184.

[2] McKeown N, Anderson T, Balakrishnan H, et al. OpenFlow: enabling innovation in campus networks[J]. ACM SIGCOMM computer communication review, 2008, 38(2): 69-74.

[3] Han B, Gopalakrishnan V, Ji L, et al. Network function virtualization: Challenges and opportunities for innovations[J]. IEEE communications magazine, 2015, 53(2): 90-97.

[4] Fei X, Liu F, Jin H, et al. FlexNFV: Flexible network service chaining with dynamic scaling[J]. IEEE Network, 2020, 34(4): 203-209.

[5] Li M, Zhang Q, Liu F. Finedge: A dynamic cost-efficient edge resource management platform for NFV network[C]//2020 IEEE/ACM 28th International Symposium on Quality of Service (IWQoS). IEEE, 2020: 1-10.

[6] Wang T, Xu H, Liu F. Multi-resource load balancing for virtual network functions[C]//2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS). IEEE, 2017: 1322-1332.

[7] J. Halpern and C. Pignataro, "Service Function Chaining (SFC) Architecture," IETF RFC 7665, 2015. [Online]. Available: https://tools.ietf.org/html/rfc7665. [Accessed: 10-Dec-2015].

[8] Zhang Q, Xiao Y, Liu F, et al. Joint optimization of chain placement and request scheduling for network function virtualization[C]//2017 IEEE 37th international conference on distributed computing systems (ICDCS). IEEE, 2017: 731-741.

[9] Jang I, Suh D, Pack S, et al. Joint optimization of service function placement and flow distribution for service function chaining[J]. IEEE Journal on Selected Areas in Communications, 2017, 35(11): 2532-2541.

[10] Pei J, Hong P, Xue K, et al. Two-phase virtual network function selection and chaining algorithm based on deep learning in SDN/NFV-enabled networks[J]. IEEE Journal on Selected Areas in Communications, 2020, 38(6): 1102-1117.

[11] Xiao Y, Zhang Q, Liu F, et al. NFVdeep: Adaptive online service function chain deployment with deep reinforcement learning[C]//Proceedings of the International Symposium on Quality of Service. 2019: 1-10.

[12] Fei X, Liu F, Xu H, et al. Towards load-balanced VNF assignment in geo-distributed NFV infrastructure[C]//2017 IEEE/ACM 25th International Symposium on Quality of Service (IWQoS). IEEE, 2017: 1-10.

[13] Jin P, Fei X, Zhang Q, et al. Latency-aware VNF chain deployment with efficient resource reuse at network edge[C]//IEEE INFOCOM 2020-IEEE Conference on Computer Communications. IEEE, 2020: 267-276.

[14] Bagaa M, Taleb T, Bernabe J B, et al. QoS and resource-aware security orchestration and life cycle management[J]. IEEE Transactions on Mobile Computing, 2020.

[15] Sciancalepore V, Samdanis K, Costa-Perez X, et al. Mobile traffic forecasting for maximizing 5G network slicing resource utilization[C]//IEEE INFOCOM 2017-IEEE Conference on Computer Communications. IEEE, 2017: 1-9.

[16] Raza M R, Fiorani M, Rostami A, et al. Dynamic slicing approach for multi-tenant 5G transport networks[J]. Journal of Optical Communications and Networking, 2018, 10(1): A77-A90.

[17] Wang G, Feng G, Quek T Q S, et al. Reconfiguration in network slicing—Optimizing the profit and performance[J]. IEEE Transactions on Network and Service Management, 2019, 16(2): 591-605.

[18] Wei F, Feng G, Sun Y, et al. Network slice reconfiguration by exploiting deep reinforcement learning with large action space[J]. IEEE Transactions on Network and Service Management, 2020, 17(4): 2197-2211.

[19] Mada B E, Bagaa M, Tale T, et al. Latency-aware service placement and live migrations in 5G and beyond mobile systems[C]//ICC 2020-2020 IEEE International Conference on Communications (ICC). IEEE, 2020: 1-6.

[20] Zhao D, Sun G, Liao D, et al. Mobile-aware service function chain migration in cloud–fog computing[J]. Future Generation Computer Systems, 2019, 96: 591-604.

[21] Addad R A, Dutra D L C, Taleb T, et al. AI-based network-aware Service Function Chain migration in 5G and beyond networks[J]. IEEE Transactions on Network and Service Management, 2021, 19(1): 472-484.

[22] Zhang Q, Liu F, Zeng C. Online Adaptive interference-aware VNF deployment and migration for 5G network slice[J]. IEEE/ACM Transactions on Networking, 2021, 29(5): 2115-2128.

[23] Mijumbi R, Serrat J, Gorricho J L, et al. Design and evaluation of algorithms for mapping and scheduling of virtual network functions[C]//Proceedings of the 2015 1st IEEE conference on network softwarization (NetSoft). IEEE, 2015: 1-9.

[24] Dolati M, Hassanpour S B, Ghaderi M, et al. DeepViNE: Virtual network embedding with deep reinforcement learning[C]//IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE, 2019: 879-885.

[25] Guan W, Wen X, Wang L, et al. A service-oriented deployment policy of end-to-end network slicing based on complex network theory[J]. IEEE access, 2018, 6: 19691-19701.

[26] Addad R A, Taleb T, Flinck H, et al. Network slice mobility in next generation mobile systems: Challenges and potential solutions[J]. IEEE Network, 2020, 34(1): 84-93.

[27] Taleb T, Ksentini A. Follow me cloud: interworking federated clouds and distributed mobile networks[J]. IEEE Network, 2013, 27(5): 12-19.

[28] Taleb T, Ksentini A, Frangoudis P A. Follow-me cloud: When cloud services follow mobile users[J]. IEEE Transactions on Cloud Computing, 2016, 7(2): 369-382.

[29] Aissioui A, Ksentini A, Gueroui A M, et al. On enabling 5G automotive systems using follow me edge-cloud concept[J]. IEEE Transactions on Vehicular Technology, 2018, 67(6): 5302-5316.

[30] Addad R A, Dutra D L C, Bagaa M, et al. Towards studying service function chain migration patterns in 5G networks and beyond[C]//2019 IEEE Global Communications Conference (GLOBECOM). IEEE, 2019: 1-6.

[31] Addad R A, Dutra D L C, Bagaa M, et al. Fast service migration in 5G trends and scenarios[J]. IEEE Network, 2020, 34(2): 92-98.

[32] Zhang J, Ye M, Guo Z, et al. CFR-RL: Traffic engineering with reinforcement learning in SDN[J]. IEEE Journal on Selected Areas in Communications, 2020, 38(10): 2249-2259.

[33] Zeng C, Liu F, Chen S, et al. Demystifying the performance interference of co-located virtual network functions[C]//IEEE INFOCOM 2018-IEEE Conference on Computer Communications. IEEE, 2018: 765-773.

[34] Ebrahimi S, Zakeri A, Akbari B, et al. Joint resource and admission management for slice-enabled networks[C]//NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium. IEEE, 2020: 1-7.

[35] R. Sedgewick, ”Algorithms in C, Part 5: Graph Algorithms”, Addison Wesley Professional, 3rd ed., 2001.

[36] Wang G, Feng G, Tan W, et al. Resource allocation for network slices in 5G with network resource pricing[C]//GLOBECOM 2017-2017 IEEE Global Communications Conference. IEEE, 2017: 1-6.

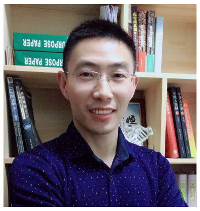
[37] Wen R, Feng G, Tang J, et al. On robustness of network slicing for next-generation mobile networks[J]. IEEE Transactions on Communications, 2018, 67(1): 430-444.

[38] Zhang N, Liu Y F, Farmanbar H, et al. Network slicing for service-oriented networks under resource constraints[J]. IEEE Journal on Selected Areas in Communications, 2017, 35(11): 2512-2521.

[39] *Networkx*. Accessed: Mar. 28, 2018. [Online]. Available: http://networkx.lanl.gov

[40] Alliance N. Description of network slicing concept[J]. NGMN 5G P, 2016, 1(1): 1-11.

Haoxian Feng received the bachelor’s degree in network engineering from Fujian Agriculture and Forestry University, Fuzhou, China, in 2020. He is he is now pursuing his B.E. degree in College of Computer Science and Technology, Fujian Agriculture and Forestry University. His research interests include software define network, network slice in 5G/6G, and network resource management based on machine learning & multi-arms bandit model.



Zhaogang Shu is currently an Associate Professor at the College of Computer and Information Science, Fujian Agriculture and Forestry University, Fuzhou, China. He also is the director of the department of computer science and Cloud Computing Lab, Fujian Agriculture and Forestry University. He received B.S. and M.S. degrees in computer science from Shantou University, China in 2002 and 2005 respectively. He also received Ph.D. degree from South China University of Technology, Guangzhou, China, in 2008. From Sept. 2008 to July 2012, he worked as a senior engineer and project manager at Ruijie Network Corporation, Fuzhou, China. From Oct. 2018 to Oct. 2019, he worked as a visiting professor at the department of communications and networking, School of Electrical Engineering, Aalto University, Finland. His research interests include software-defined network, network function virtualization, 5G network, network security, machine learning and cloud computing. He serves as the reviewers of many famous journals on network technology, including IEEE Network, IEEE/ACM Transactions on Networking, ACM/Springer Mobile Networks and Applications. He directed more than 8 research projects and was the author of 30 papers and 5 patents.

Tarik Taleb is currently a Professor at the Center of Wireless Communications, The University of Oulu, Finland. He is the founder and director of the MOSA!C Lab (www.mosaic-lab.org). Between Oct. 2014 and Dec. 2021, he was a Professor at the School of Electrical Engineering, Aalto University, Finland. Prior to that, he was working as Senior Researcher and 3GPP Standards Expert at NEC Europe Ltd, Heidelberg, Germany. Before joining NEC and till Mar. 2009, he worked as assistant professor at the Graduate School of Information Sciences, Tohoku University, Japan, in a lab fully funded by KDDI. From Oct. 2005 till Mar. 2006, he worked as research fellow at the Intelligent Cosmos Research Institute, Sendai, Japan. He received his B. E. degree in Information Engineering with distinction, M.Sc. and Ph.D. degrees in Information Sciences from Tohoku Univ., in 2001, 2003, and 2005, respectively.

Prof. Taleb’s research interests lie in the field of telco cloud, network softwarization & network slicing, AI-based software defined security, immersive communications, mobile multimedia streaming, and next generation mobile networking. Prof. Taleb has been also directly engaged in the development and standardization of the Evolved Packet System as a member of 3GPP’s System Architecture working group 2. Prof. Taleb served on the IEEE Communications Society Standardization Program Development Board.

Prof. Taleb served as the general chair of the 2019 edition of the IEEE Wireless Communications and Networking Conference (WCNC’19) held in Marrakech, Morocco. He was the guest editor in chief of the IEEE JSAC Series on Network Softwarization & Enablers. He served on the editorial board of the IEEE Transactions on Wireless Communications, IEEE Wireless Communications Magazine, IEEE Journal on Internet of Things, IEEE Transactions on Vehicular Technology, IEEE Communications Surveys & Tutorials, and a number of Wiley journals. Till Dec. 2016, he served as chair of the Wireless Communications Technical Committee, the largest in IEEE ComSoC.

Prof. Taleb is the recipient of the 2021 IEEE ComSoc Wireless Communications Technical Committee Recognition Award (Dec. 2021), the 2017 IEEE ComSoc Communications Software Technical Achievement Award (Dec. 2017) for his outstanding contributions to network softwarization. He is also the (co-) recipient of the 2017 IEEE Communications Society Fred W. Ellersick Prize (May 2017), and many other awards from Japan. Some of Prof. Taleb’s research work have been also awarded best paper awards at prestigious IEEE-flagged conferences.

Yuantao Wang received the bachelor's degree in information management and information system from Anqing Normal University, Anqing, China, in 2021. She is currently pursuing the master's degree at Computer Science and Technology, Fujian Agriculture And Forestry University. Her research interests include software defined networking and network functional virtualization. She is now undertaking the research work on reducing the operating expense of network operators with the constrained optimization.

Zhiwei Liu is currently pursuing a master's degree at Fujian Agriculture and Forestry University. His research interests include deep reinforcement learning, network functions virtualization, and cloud computing. He is now conducting research work on the scheduling of VNFs through deep reinforcement learning.