Joint Availability Guarantee and Resource Optimization of Virtual Network Function Placement in Data Center Networks

Meng Wang, *Student Member, IEEE*, Bo Cheng, *Member, IEEE*, and Junliang Chen

***Abstract*—Network Function Virtualization (NFV) is a promising technology that decouples network functions from the physical device on which they deployed. Network service in NFV is deployed as Service Function Chain (SFC) that consists of an ordered set of Virtual Network Functions (VNFs). In this paper, we focus on the VNF placement problem in data center networks considering availability guarantee and resource optimization. Firstly, we define an availability model that takes both physical device failures and VNF failures into consideration when evaluating the availability of SFC. Secondly, we propose a novel Joint Path-VNF (JPV) backup model that combines path backup and VNF backup in a joint way. In the JPV backup model, resource consumption can be effectively reduced. Finally, we design an Affinity-Based Algorithm (ABA) to reduce physical link consumption when map VNFs. The evaluation results show that ABA and JPV can achieve better availability improvement (99.99%) with less resource (reduce 40% PL consumption).**

***Index Terms*—Network function virtualization, service function chain, availability, backup, resource allocation.**

I. Introduction

NETWORK Function Virtualization (NFV) [1] is a promising technology that decouples network functions from the hardware. In this architecture, network functions or middleboxes traditionally attached on dedicated hardware are now realized in software that can run on any Commercial Off The Shelf (COTS) servers. Due to the convenient and flexible management of Virtual Network Functions (VNFs), NFV significantly reduces the Capital Expenditure (CAPEX) and Operating Expense (OPEX) and plays an important role in communication networks, such as mobile networks, enterprise networks, and data center networks.

As for the network service in NFV, it includes an ordered set of VNFs, generally referred to as an SFC (Service Function Chain) [2]. The SFC needs to be placed in the cloud data center and run on the physical device. Availability requirements are important issues that need to be carefully considered. Compared with the traditional IT applications that require two 9s to three 9s (i.e., 99% and 99.9%) [3], the telecom network services often require five 9s or even six 9s. However, the failures of both physical device and VNF bring new challenges in providing SFC with availability guarantee. The availability of SFC can be guaranteed through the redundancy technology [4]. In this paper, we use the active/standby redundancy model [5], [6], in which a standby entity can be used if a VNF fails. However, resource consumption increases as redundancy increases. It is necessary to reduce resource consumption when providing backup for VNFs. Therefore, VNF placement becomes an important but difficult problem, which has received increasing attention from both academia [7], [8] and industry [9], [10].

There are different kinds of backup models and placement algorithms in the existing work to improve the availability of SFC and map VNFs. However, there are still some problems remaining to be solved:

- The failures of physical device, such as PMs (Physical Machines) and switches, are often ignored. The failures of VNF may affect the availability of SFC. And the failures of physical device may also result in the unavailability of several VNFs in one SFC. Therefore, we have to consider the failures of both physical device and VNF [11].
- Traditionally, path backup and VNF backup are often considered separately. Providing only backup paths or providing only backup VNFs cannot meet the availability requirements of SFC. When the one achieves the highest performance, the other will become a bottleneck for availability improvement.
- Most of the existing work pays more attention to improving the availability level of SFC. However, as the backup model increases the availability of SFC, resource consumption also increases. Besides, the VNF placement problem aims at optimizing resource consumption, becomes a critical research issue.

Given these facts, we focus on the VNF placement problem in data center networks considering availability guarantee and resource optimization. We take the failures of both physical device and VNF into account when evaluating the availability of SFC. And we propose a novel Joint Path-VNF (JPV) backup model that considers both path backup and VNF backup. It

Manuscript received August 23, 2019; revised December 20, 2019; accepted March 3, 2020. Date of publication March 6, 2020; date of current version June 10, 2020. This work is supported by the National Key Research and Development Program of China under grant 2018YFB1003804, in part by the National Natural Science Foundation of China under grant 61921003, 61972043, and in part by the BUPT Excellent Ph.D. Students Foundation under grant CX2019214. The associate editor coordinating the review of this article and approving it for publication was H. Lutfiyya. *(Corresponding author: Bo Cheng.)*

The authors are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: mengwang@bupt.edu.cn; chengbo@bupt.edu.cn; chjl@bupt.edu.cn).

Digital Object Identifier 10.1109/TNSM.2020.2978910

is worth mentioning that JPV is not just a simple combination of path backup and VNF backup. We design the model in a joint way where one PM can provide backup for two PMs. In this way, the number of PLs (physical links), PMs (physical machines) and VNFs can be reduced effectively. The VNF placement problem is known to be NP-hard [12], [13], so as the VNF placement problem in this paper. Therefore, we propose an Affinity-Based Algorithm (ABA) to solve this problem. According to the number of connection PLs between PMs in the data center networks, the ABA divides PMs into different levels of affinity groups. By mapping adjacent VNFs in the same affinity group, communication overhead can be effectively reduced.

In summary, the main contributions of this paper are summarized as follows:

- We define an availability model that takes both physical device failures and VNF failures into account. In this model, we consider the failures of physical device such as PMs and switches. Although it can make our problem harder, it closer to reality.
- We propose a novel JPV backup model that jointly considers both path backup and VNF backup. In this model, the backup PM which is placed on the backup path provides backup for two PMs simultaneously. By providing the sharable backup, JPV model can effectively reduce the number of PLs, PMs, and VNFs used.
- We design an Affinity-Based Algorithm (ABA) to reduce PL consumption. The ABA divides PMs into different levels of affinity groups according to the communication overhead of PMs. The VNFs are mapped based on the affinity groups, thereby reducing PL consumption.

The remainder of this paper is organized as follows. In Section II, we review and discuss the related work. Then the system model is presented in Section III. In Section IV, we describe and formulate the VNF placement problem. Our proposed algorithm is described in Section V. Section VI evaluates the performance of our proposed solutions. Finally, we conclude this study in Section VII.

II. Related Work

In this paper, we focus on the VNF placement problem considering availability guarantee and resource optimization. Therefore, we focus on the three aspects related to the VNF placement problem, including the availability model, backup model, and placement algorithm.

Availability Model: The first thing we need in availability guarantee is the availability model. In [14]–[16], the authors define an optimal VNF placement problem. They define an availability model that considers VNF failures. And they present an algorithm that minimizes resource consumption while guaranteeing the availability of SFC. Qu *et al.* [17] define an availability model considers VNF failures and optimize traffic routing in data center networks for the VNF placement problem. Herker *et al.* [18] mention physical device failures in data center networks, but do not provide a clear availability model that considers both VNF failures and physical device failures. Sun *et al.* [19] proposed an algorithm to ensure reliability while taking node failures and link failures into consideration.

We can conclude that most of the existing work focuses only on VNF failures in the availability model. Different from the traditional work, we define an availability model that takes both physical device failures and VNF failures into consideration. In our proposed availability, it closer to the real data center networks.

Backup Model: Most of the existing work improves availability by proposing backup models and algorithms. Kang *et al.* [20] study the tradeoff between computational due and reliability for VNF. Kong *et al.* [21] propose a protection mechanism that combines both VNF replicas and backup path protection to guarantee the availability of SFC. They combine path backup and VNF backup for the first time, but it is just a simple combination of the two backup methods. In [14], the authors propose a joint protection model that uses one physical node to provide backup for two physical nodes. However, the joint protection model only focuses on path backup rather than combined path backup and VNF backup. Besides, the joint protection model requires more PLs to enable VNFs to communicate with each other.

Different from the traditional backup models (e.g., Path backup model, VNF backup model, Path-VNF backup model, and Joint Protection backup model), we propose a novel backup model that combines path backup and VNF backup in a joint way. In our proposed backup model, one PM provides backup for two PMs simultaneously. Therefore, the proposed backup model can achieve higher availability with less resource consumption.

Placement Algorithm: Recently, there are multiple algorithms designed to solve the VNF placement problem in data center networks. Luizelli *et al.* [22] formulate the VNF placement problem and propose an ILP (Integer Linear Programming) model to solve it. Rankothge *et al.* [23] propose a genetic algorithm to solve the resource allocation algorithm for VNFs. And they [12] present a comprehensive analysis of two genetic algorithms. Ye *et al.* [24] propose solutions to optimize network configuration and ensure availability. This approach primarily optimizes topology design rather than reducing resource consumption. Herker *et al.* [18] model different backup strategies and evaluate them in different data center topologies. For a given SFC request, they can answer which data center topology is the most appropriate one to deploy. However, they are not concerned about how to reduce resource consumption in a certain data center.

We can observe that most of the existing placement algorithms focus on the VNF placement problem without availability requirements. In our proposed solutions, we map VNFs considering not only resource optimization but also availability guarantee in data center networks.

In summary, most of the existing work focuses on the availability model, backup model, and placement algorithm. However, traditional solutions only consider VNF failures and often ignore physical device failures. And existing work cannot achieve high availability while optimizing resource consumption. In this paper, we firstly define an availability model

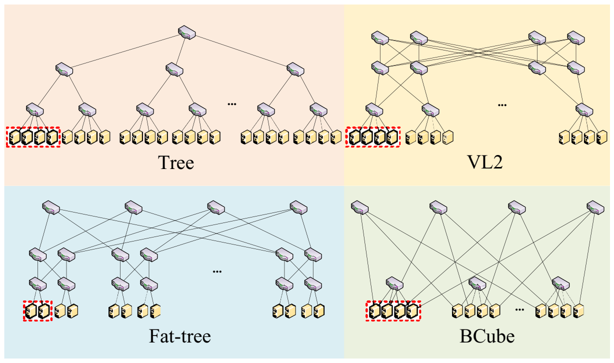


Fig. 1. Data center network topology.

to take both physical device failures and VNF failures into consideration. Secondly, we propose JPV backup model to guarantee availability while reducing resource consumption. Finally, we design an affinity-based algorithm to efficiently reduce PL consumption.

III. System Model

In this section, we describe the system model. Firstly, we introduce the data center network model consisting of four typical types of architectures. Secondly, we define an availability model to take both physical device failures and VNF failures into consideration. Thirdly, we present five backup models and discuss them in terms of resource optimization and availability improvement. Finally, we propose an affinity model used in the placement algorithm.

A. Data Center Network Topology

As Fig. 1 shows, we present four typical types of data center architectures including Tree, VL2, Fat-tree, and BCube, which are described in detail in [25].

It is worth mentioning that in the data center architecture, the PMs can be divided into multiple domains (as the red dotted line in Fig. 1 shows). The communication overheads between different domains are different, which we will elaborate on the affinity model.

B. Availability Model

In this subsection, we divide the availability into single VNF (or physical device) availability, component availability and SFC availability. A component contains physical device and VNF. And an SFC is composed of several components.

1) Availability of Single VNF (or Physical Device): The availability is the probability that the system can work properly during a certain time. Therefore, the status of VNF and physical device can be divided into Uptime and Downtime, which can be characterized in terms of Mean Time Between Failures (MTBF) and Mean Time To Repair (MTTR), respectively. In this paper, we assume that the failure of each VNF or each physical device is independent. Therefore, the availability of VNF (or physical device) can be characterized as follows:

TABLE I
MTBF for Physical Device in Data Center

Physical Device	**MTBF (hours)**	**MTTR (hours)**
Server	0.6667×10^4 - 10.95×10^4	7
ToR switch	14.5×10^4 - 17.52×10^4	2.9
Aggregation switch	8.76×10^4 - 20×10^4	2.1
Core switch	60×10^4	2.1

$$A_{v_p}(A_{n_i}) = \frac{Uptime}{Uptime + Downtime} = \frac{MTBF}{MTBF + MTTR} \tag{1}$$

where A_{v_p} (or A_{n_i}) indicates the availability of VNF (or physical device).

As Table I shows, we use the three-year failure event logs [5], [26] in the data center to calculate MTBF and MTTR. The physical device we focus on includes Intel servers and Cisco switches.

2) Availability of Component: Network service in NFV can be accomplished by SFC, which consists of an ordered set of VNFs. Therefore, we treat the network service request as an SFC Request (SFCR).

We use $\mathbb{SR}$ to indicate the set of SFCRs. Therefore, $\mathbb{SR}^s = (N^s, L^s, A^s)$ indicates the SFCR s. N^s indicates the set of VNFs in s. And L^s indicates the logical links that used to connect VNFs. According to the characteristics of network service, the SFCR has availability requirements, which can be indicated by A^s.

Considering the availability of PM, we treat the PM and VNF running in it as a component c_i. Thus, the availability of the component is:

$$A_{c_i} = A_{n_i} A_{v_p} \tag{2}$$

where A_{c_i} is the availability of component i. A_{n_i} is the availability of PM_i, and A_{v_p} is the availability of VNF_p which is running in PM_i.

As mentioned above, an SFC is composed of an ordered set of components. The components of SFC are organized in a sequential or parallel manner. We use A_{seq} or A_{para} to indicate the availability of two components. A_{c_i} is the availability of component i, so is A_{c_j}. Then, the availability of the two components are as follows:

1) *Sequential:* As for the sequential manner, each component has to be available at the same time. Therefore, the availability of the two components is:

$$A_{seq} = A_{c_i} A_{c_j} \tag{3}$$

2) *Parallel:* The parallel way can be seen as the parallel of the working path and the backup path. Network traffic is forwarded in the working path [27]. At least one of the two paths is available to guarantee the overall availability. Thus, the availability of the two components is:

$$\begin{aligned} A_{para} &= 1 - (1 - A_{c_i})(1 - A_{c_j}) \\ &= A_{c_i} + A_{c_j} - A_{c_i} A_{c_j} \end{aligned} \tag{4}$$

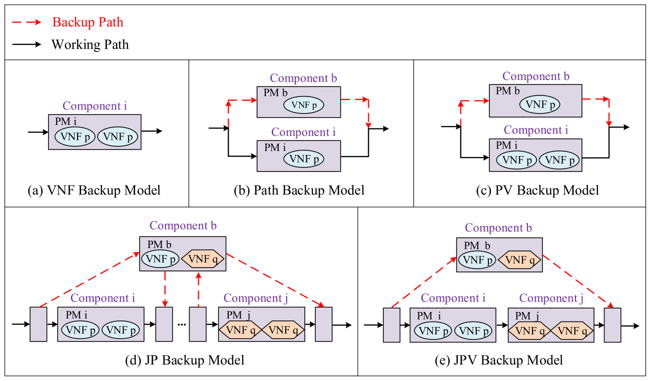


Fig. 2. Backup models: (a) Path; (b) VNF; (c) PV; (d) JP; (e) JPV.

C. Availability of SFC

In this paper, we define that an SFC consists of several components. Therefore, using the basic availability models (*Eqs.* (2), (3) and (4)) mentioned above, we can evaluate the availability of SFC:

$$A_{sfc} = \left(\prod_{c_i, c_j \in seq} A_{seq} \right) \left(\prod_{c_i, c_j \in para} A_{para} \right) \tag{5}$$

where A_{sfc} indicates the availability of SFC.

D. Backup Model

In this section, we discuss five backup models. Firstly, we describe the VNF and Path backup models. Then we describe the PV, JP and JPV backup models that aim at combing path backup and VNF backup. Finally, we summarize the availability improvement and resource consumption of the five backup models.

We use A_{sfc} to indicate the availability of SFC. A_{n_i}, A_{n_j} and A_{n_b} indicate the availability of PM_i, PM_j and backup PM_b. And A_{v_p}, A_{v_q} are used to indicate the availability of VNF_p and VNF_q. θ_b indicates the number of redundant VNFs in PM_b, and so is θ_i and θ_j.

In this paper, we assume that the availability of VNF_p and VNF_q is independent.

1) VNF Backup Model: Fig. 2(a) shows two VNF_p running in PM_i. The availability of SFC can be guaranteed by ensuring that PM is available and at least one of the VNFs in PM is available. Therefore, the availability of SFC is:

$$A_{sfc} = \prod \left(A_{n_i} \left(1 - \left(1 - A_{v_p} \right)^{\theta_i} \right) \right) < \prod A_{n_i} \tag{6}$$

The VNF backup model adds backup VNF to improve the availability of SFC. There are only one VNF cost and no PL cost in one component. And we can observe that the availability of SFC is limited by the availability of PM.

2) Path Backup Model: As Fig. 2(b) shows, VNF_p is in the backup path while another VNF_p is in the working path. The availability of SFC can be guaranteed by ensuring that at least one of the path is available. For example, the working path is available which means that the PM and VNF in the working path are all available. Thus, the availability of SFC is:

$$\begin{aligned} A_b &= A_{n_b} A_{v_p}, A_i = A_{n_i} A_{v_p} \\ A_{sfc} &= \prod \left(A_{c_b} + A_{c_i} - A_{c_b} A_{c_i} \right) < \prod A_{v_p} \end{aligned} \tag{7}$$

In this model, each component adds a backup path. The extra resource used contains one VNF and two PLs. We can conclude that the availability of VNF is the bottleneck.

3) PV Backup Model: From Eq. (6), we can conclude that the availability of PM is the bottleneck in VNF backup model. And in Eq. (7), the availability of VNF is the bottleneck.

Therefore, we describe PV (Path-VNF) backup model [21], as shown in Fig. 2(c). It is a combination of VNF backup model and Path backup model. One VNF_p is in the backup path and another two VNF_p are in the working path. The SFC is available while at least one of the path is available. Therefore, the availability of SFC is:

$$\begin{aligned} A_b &= A_{n_b} \left(1 - \left(1 - A_{v_p} \right)^{\theta_b} \right) \\ A_i &= A_{n_i} \left(1 - \left(1 - A_{v_p} \right)^{\theta_i} \right) \\ A_{sfc} &= \prod \left(A_b + A_i - A_b A_i \right) \end{aligned} \tag{8}$$

where A_b indicates both PM_b and VNF_p running in it are available, so is A_i. In this model, we set θ_b to 1 by default.

The PV backup model has a higher availability improvement than VNF backup model and Path backup model. However, each component adds two PLs and two VNFs, which is more than the two backup models. With this in mind, we propose a novel JPV backup model.

4) JP Backup Model: The JP (Joint Protection [14]) backup model is initially focused on path backup. In this article, we extend the JP backup model to the path and VNF backup, as shown in Fig. 2(d).

The difference between JP and JPV is that the former provides backup for two non-adjacent PMs, and the latter provides backup for two adjacent PMs. As for the availability of SFC, the calculation method of JP backup model is the same as JPV (Joint Path-VNF) backup model, which we will elaborate later.

For resource consumption, a component of JP adds two VNFs and two PLs on average, as shown in Fig. 2(d).

5) JPV Backup Model: As Fig. 2(e) shows, we propose a novel JPV (Joint Path-VNF) backup model to optimize resource consumption. In this model, we consider three components, working component *i*, *j* and backup component *b*.

The availability of SFC can be guaranteed by ensuring that at least one of the path is available. Therefore, the availability of SFC is:

$$\begin{aligned} A_{c_b} &= A_{n_b} A_{v_p} A_{v_q} \\ A_{c_i} &= A_{n_i} \left(1 - \left(1 - A_{v_p} \right)^{\theta_i} \right) \\ A_{c_j} &= A_{n_j} \left(1 - \left(1 - A_{v_q} \right)^{\theta_j} \right) \\ A_{sfc} &= \prod \left(A_{c_b} + A_{c_i} A_{c_j} - A_{c_b} A_{c_i} A_{c_j} \right) \end{aligned} \tag{9}$$

where A_{c_b} indicates PM_b, VNF_p and VNF_q running in the backup component are all available, and so is A_{c_i} and A_{c_j}.

TABLE II
AVERAGE RESOURCE CONSUMPTION OF BACKUP MODELS

Backup Model	VNF Consumption (#)	PL Consumption (#)
VNF	1	0
Path	1	2
PV	2	2
JP	2	2
JPV	2	1

TABLE III
PARAMETER VALUES

Parameter	Value
$A_{v_p} = A_{v_q}$	$[0.99, 0.9999]$
$A_{n_b} = A_{n_i} = A_{n_j}$	$[0.999, 0.99999]$
$x = A_b$	$0.99999 * 0.9999 * 0.9999$
$y = A_i = A_j$	$0.999 * (1 - (1 - 0.99)^2)$
Δ	$-2.54 * 10^{-10}$

In our proposed JPV backup model, a component adds two VNFs and one PL on average, which is better than the JP and PV backup models.

6) Discussion: In this subsection, we discuss the five backup models in terms of resource consumption and availability improvement.

In summary, Table II presents the average resource consumption of the five backup models. We can conclude that VNF and Path backup model have less resource consumption. However, the availability of VNF and PM is the bottleneck in the two backup models. Considering the availability improvement, we focus on the PV, JP, and JPV backup models. We can observe that our proposed JPV has less resource consumption than PV and JP.

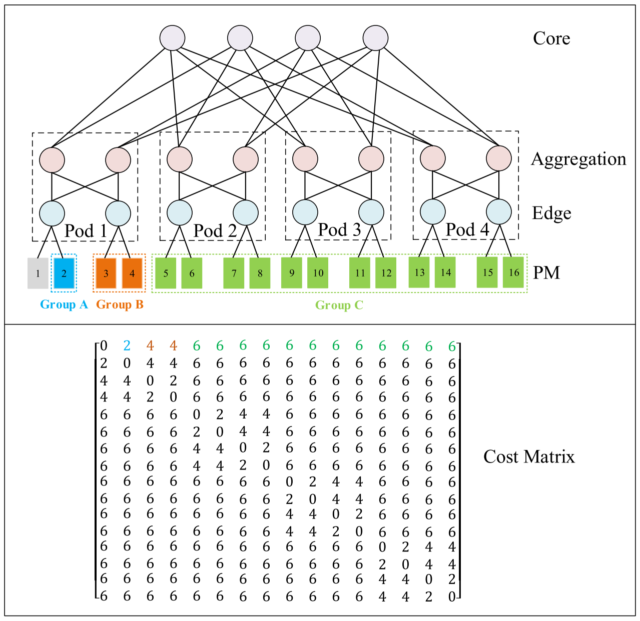
In general, we can reduce PL consumption by using one PM to provide backup for three or more adjacent PMs. However, it requires the backup PM to hold more VNFs (three or even more). If not, it will make the backup model more complicated. More importantly, availability improvement will be reduced. Therefore, in this paper, we use one PM to provide backup for two adjacent PMs.

Then, we analyze the availability of PV (Eq. (8)) and JPV (Eq. (9)). For simplicity, we assume A_b = x, $A_i = A_j$ = y in Eqs. (8) and (9). We consider the availability of two components. In this case, PV backup model needs to calculate Eq. (8) twice, and JPV backup model only needs to calculate once. Therefore, the square of Eq. (8) minus Eq. (9) is:

$$\begin{aligned}\Delta &= (x + y - xy)^2 - \left(y^2 + x - xy^2\right)\\ &= \left(x(x-1)(y-1)^2\right) \end{aligned} \tag{10}$$

From Eq. (10), we can conclude that when x is the maximum value and y is the minimum value, Δ achieves the maximum value. The parameter values (detailed in Section VI-B) can be found in Table III.

Δ is so small (at the level of 10^{-10}) that can be ignored under four 9s requirements. Therefore, we can think that PV and JPV are close in terms of improving availability, and we will elaborate in Section VI.



$$\begin{bmatrix}
0 & 2 & 4 & 4 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6\\
2 & 0 & 4 & 4 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6\\
4 & 4 & 0 & 2 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6\\
4 & 4 & 2 & 0 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6\\
6 & 6 & 6 & 6 & 0 & 2 & 4 & 4 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6\\
6 & 6 & 6 & 6 & 2 & 0 & 4 & 4 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6\\
6 & 6 & 6 & 6 & 4 & 4 & 0 & 2 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6\\
6 & 6 & 6 & 6 & 4 & 4 & 2 & 0 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6\\
6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 0 & 2 & 4 & 4 & 6 & 6 & 6 & 6\\
6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 2 & 0 & 4 & 4 & 6 & 6 & 6 & 6\\
6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 4 & 4 & 0 & 2 & 6 & 6 & 6 & 6\\
6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 4 & 4 & 2 & 0 & 6 & 6 & 6 & 6\\
6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 0 & 2 & 4 & 4\\
6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 2 & 0 & 4 & 4\\
6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 4 & 4 & 0 & 2\\
6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 4 & 4 & 2 & 0
\end{bmatrix}$$

Fig. 3. Fat-tree topology and cost matrix.

E. Affinity Model

As Fig. 3 shows, we use a 3-layer fat-tree [28] data center topology. The fat-tree topology contains three types of switches including core switch, aggregation switch, and edge switch. In a k-ray fat-tree, there are k pods, with $k/2$ aggregation switches and $k/2$ edge switches in each pod. And each pod is connected with $(k/2)^2$ core switches and with $(k/2)^2$ PMs.

In the data center networks, we use $\mathbb{PN}^r = (N^r, L^r, S^r)$ to describe the physical network r. N^r indicates the set of PMs in physical network. L^r indicates all of the PLs and S^r indicates the set of total switches.

In previous work [29], VNF is mapped based on affinity and anti-affinity constraints. In this paper, we propose the affinity group, which consists of PMs with the same communication overhead.

In the fat-tree topology, we get the cost matrix based on the number of PLs between PMs, as shown in Fig. 3. For example, the number of PLs between PM1 and PM2 is 2, that is, $M_{12} = 2$. M can be shown as:

$$M_{ij} = \begin{cases} 0 & \text{if } i = j \\ 2 & \text{if } \left\lceil \frac{2(i-1)}{k} \right\rceil = \left\lceil \frac{2(j-1)}{k} \right\rceil \\ 4 & \text{if } \left\lceil \frac{2(i-1)}{k} \right\rceil \neq \left\lceil \frac{2(j-1)}{k} \right\rceil \wedge \left\lfloor \frac{4(i-1)}{k^2} \right\rfloor = \left\lfloor \frac{4(j-1)}{k^2} \right\rfloor \\ 6 & \text{if } \left\lceil \frac{4(i-1)}{k^2} \right\rceil \neq \left\lceil \frac{4(j-1)}{k^2} \right\rceil \end{cases} \tag{11}$$

where i, j are the IDs of PMs. The cost matrix M is a function of k, the total number of ports on each switch. For example, we compute M_{12} (i = 1 and j = 2). k is the number of ports on each switch (k = 4 in a 4-ray fat-tree). Therefore, we can conclude that $M_{12} = 2$ according to Eq. (11).

TABLE IV
AFFINITY GROUPS OF PM1

Group	Cost	PMs
A	2	PM2
B	4	PM3, PM4
C	6	PM5, PM6, PM7, PM8, PM9, PM10, PM11, PM12, PM13, PM14, PM15, PM16

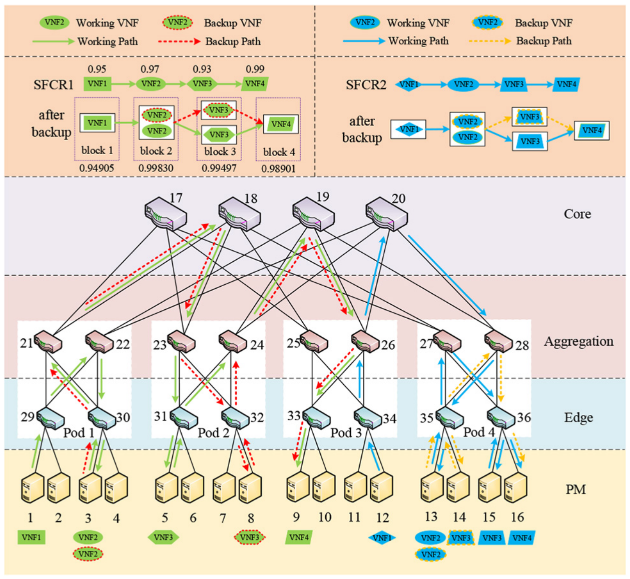


Fig. 4. VNF placement in fat-tree topology.

Based on these features, we divide the PMs into different levels of affinity groups. For example, we present the affinity groups of PM1, as shown in Table IV.

Table IV shows the three affinity groups of PM1 with different costs. Group A consisting of PM2 is the first affinity group with a cost value of 2. And Group B consisting of PM3 and PM4 is the second affinity group with a cost value of 4. Group C consisting of the remaining PMs is the third affinity group with a cost value of 6.

Taking into account the affinity groups, two connected VNFs can be placed in the same affinity group with low-cost value as much as possible during the VNF placement process. In this way, the PL consumption can be efficiently reduced.

IV. Problem Statement

In this section, we first provide a simple problem example to clarify the concepts such as availability model, backup model, placement method and resource consumption. The second part of the section is devoted to the mathematical formulation of the placement problem.

A. Problem Description

Fig. 4 shows two SFC requests (SFCRs). The availability of VNFs in SFCR1 are 0.95, 0.97, 0.93 and 0.99, which are selected randomly for this example. The availability of SFC1 is 0.84843 when the availability of PM is ignored. However, the availability of SFC is 0.84504 when the availability of PM is assumed to be 0.999 (based on *Eqs.* (2) and (3)). Therefore, the availability of PM cannot be ignored.

In the SFC1, we provide backup VNF for VNF2 and backup path for VNF3. We divide SFC1 into four blocks. The availability of each block is 0.94905, 0.99830, 0.99497 and 0.98901, respectively. Therefore, the availability of SFC1 after backup is 0.93231. Compared with 0.84504 before backup, it is a big improvement. Thus, a suitable backup model can improve availability effectively.

In the fat-tree topology of Fig. 4, VNFs are placed in PMs. For example, VNF1 in SFC1 is placed in PM1. The green solid line is the working path of SFC1, that is, 1-29-22-30-3-30-21-18-23-31-5-31-24-19-26-33-9. The red dotted line is the backup path: 3-30-21-18-23-32-8-32-24-19-26-33-9. Therefore, the number of PLs in the working path is 16. And the backup path is 12. The number of PLs can also be represented by the cost matrix, that is, $C_{13} + C_{35} + C_{59} = 16$. In the cost matrix, 1, 3, 5 and 9 are the IDs of the PMs.

It is worth mentioning the path selection (such as the PLs in C_{13}), we choose the path by introducing the availability of switch. We choose the switch with the highest availability among switches that can be selected. For example, we can connect Switch21 or Switch22 with Switch29. Because the availability of Switch22 is higher, we choose to connect Switch22 and Switch29.

Similarly, the blue solid line is the working path in SFC2 while the orange dotted line is the backup path. The working path in SFC2 is 12-34-26-20-28-35-13-35-27-36-15-36-16, and the backup path is 13-35-14-35-28-36-16. Therefore, the number of PLs in the working path is 12 and the backup path is 6. From this case, we can conclude that different backup models and placement methods can result in different resource consumption.

The above case shows the VNF placement problem we are concerned about. Firstly, we take the failures of both physical device and VNF into consideration. Secondly, we adopt the backup models mentioned in Section III-D to meet the availability requirements. Finally, we use ABA to map VNFs in the PMs.

In this paper, the components of the working path in SFC are first mapped to the physical network based on the affinity model. After that, the backup path is deployed based on the backup model and affinity model to guarantee availability and reduce resource optimization.

B. Problem Formulation

In this subsection, we formulate the VNF placement problem in data center networks. Table V shows the basic notations used in this paper.

In this paper, we focus on availability guarantee and resource consumption. Thus, we first meet the availability requirements of SFCR. As we mentioned in Section III, the SFC is divided into several components in the backup model. Therefore, the availability constraints are:

$$\prod_{i=1}^{|C^r|} A^s_{c_i} \geq A^s \tag{12}$$

TABLE V
BASIC NOTATIONS USED THROUGHOUT THIS PAPER

Symbol	Definition
Network	
$N^r = \{n_1^r, n_2^r \ldots\}$	the set of PMs in network r.
$L^r = \{l_1^r, l_2^r \ldots\}$	the set of PLs in network r.
$C^r = \{c_1^r, c_2^r \ldots\}$	the set of components in network r.
n_i^r, n_j^r	two PMs in network r.
$l_{ij}^r = (n_i^r, n_j^r)$	the PLs between n_i^r and n_j^r.
SFC	
$\mathbb{SR}$	the set of SFCRs, $s \in \mathbb{SR}$ is an SFCR.
$V^s = \{v_1^s, v_2^s \ldots\}$	the set of VNFs in SFC s.
$L^s = \{l_1^s, l_2^s \ldots\}$	the set of logical links in SFC s.
v_p^s, v_q^s	two VNFs in SFC s.
$l_{pq}^s = (v_p^s, v_q^s)$	the logical links between v_p^s and v_q^s.
A^s	the availability requirements of SFC s.
$A_{c_i}^s$	the availability of component c_i in SFC s.
Resource	
$cpu_{v_p^s}$	the CPU consumption of v_p^s.
$mem_{v_p^s}$	the memory consumption of v_p^s.
$bw_{(v_p^s, v_q^s)}$	the bandwidth consumption between v_p^s and v_q^s.
$C_{n_i^r}^{cpu}$	the CPU capacity of n_i^r.
$C_{n_i^r}^{mem}$	the memory capacity of n_i^r.
$C_{(n_i^r, n_j^r)}^{bw}$	the bandwidth capacity between n_i^r and n_j^r.
Matrices	
$D_{(n_i^r, n_j^r)}$	the connection matrix.
$M_{(n_i^r, n_j^r)}$	the cost matrix.
Variables	
$\alpha_{v_p^s \to n_i^r}$	whether v_p^s mapped in n_i^r.
$\beta_{l_{pq}^s \to l_{ij}^r}$	whether logical link l_{pq}^s mapped in PL l_{ij}^r.

Then we describe the resource consumption. First, we use $\alpha_{v_p^s \to n_i^r}$ to describe whether VNF v_p^s is mapped in PM n_i^r:

$$\alpha_{v_p^s \to n_i^r} = \begin{cases} 1 & \text{if } v_p^s \text{ is mapped in } n_i^r \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

We take CPU, memory and bandwidth constraints into consideration. In detail, the resource requirements of all VNFs placed in the same PM cannot exceed the resource capacity of this PM. So the constraints of CPU are:

$$\sum_{s=1}^{|\mathbb{SR}|} \sum_{v_p=1}^{|V^s|} cpu_{v_p^s} \cdot \alpha_{v_p^s \to n_i^r} \le C_{n_i^r}^{cpu} \tag{14}$$

Similar to CPU, we formulate the memory constraints as:

$$\sum_{s=1}^{|\mathbb{SR}|} \sum_{v_p=1}^{|V^s|} mem_{v_p^s} \cdot \alpha_{v_p^s \to n_i^r} \le C_{n_i^r}^{mem} \tag{15}$$

Then we use $\beta_{l_{pq}^s \to l_{ij}^r}$ to describe that whether logical link l_{pq}^s is mapped in PL l_{ij}^r:

$$\beta_{l_{pq}^s \to l_{ij}^r} = \begin{cases} 1 & \text{if } l_{pq}^s \text{ is mapped in } l_{ij}^r \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

The consumption of the PL is mainly bandwidth, so the bandwidth consumption of the logical links placed on the same PL cannot exceed the capacity of this PL. Thus, the bandwidth constraints are:

$$\sum_{s=1}^{|\mathbb{SR}|} \sum_{l_{pq}=1}^{|L^s|} bw_{l_{pq}^s} \cdot \beta_{l_{pq}^s \to l_{ij}^r} \le C_{l_{ij}^r}^{bw}. \tag{17}$$

C. Objectives

In this paper, we focus on resource optimization such as PM and PL consumption. As we discussed in Section IV-A, different backup models and placement algorithms can result in different resource consumption, especially PL consumption. Therefore, our target is to reduce PL consumption, indicated as PL cost.

We define a connection matrix D to describe whether two PMs (such as n_i^r and n_j^r) running VNFs are connected. D is as follows:

$$D_{ij} = \begin{cases} 1 & \text{if } n_i^r \text{ is connected to } n_j^r \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

In Section III-E, we define a cost matrix M to indicate the number of PLs between two PMs. Therefore, the PL cost between two PMs can be indicated as:

$$\begin{aligned} &\min \ sum(DM), \\ &s.t. \ Eq. (12) \ to \ Eq. (18). \end{aligned} \tag{19}$$

where the *sum* function is used to calculate the sum of all the elements of the matrix.

In this paper, we aim to find a trade-off solution between availability guarantee and resource optimization. In Section III-D, we propose the JPV backup model to improve availability. However, PM and PL costs increase as availability increases. Therefore, we propose ABA (Affinity-Based Algorithm) to reduce PL cost.

V. Proposed Algorithm

In this section, we describe the ABA, which consists of two stages: the mapping stage and the backup stage.

A. Framework of ABA

As we mentioned before, the SFC consists of an ordered set of VNFs. In our proposed algorithm, we sequentially map each VNF in the SFCRs. In ABA, there is a mapping stage and a backup stage. The former is responsible for mapping the working SFC based on affinity groups. The latter aims to provide backup for the working SFC based on the availability requirements and backup models.

Firstly, in the mapping stage, we map VNFs to the physical network. We treat SFCRs one at a time as they arrive. After mapping an SFCR, the algorithm will update the current network status. Then start processing the next SFCR. We traverse all VNFs in SFC until all of them are mapped to the physical network. For each VNF, we check whether the PM can hold this VNF. If so, map VNF to this PM. Otherwise, find another PM for this VNF. Then, we can map SFC to the

Algorithm 1: Mapping Stage

```
Input: The SFC request s: $\mathbb{SR}^s$ = ($N^s$, $L^s$, $A^s$);
       The physical network r: $\mathbb{PN}^r$ = ($N^r$, $L^r$, $S^r$);
Output: The mapping result: $MS_{res}$;
1  Initialize: The initial mapping data: $MS_{res} = \varnothing$;
   The initial physical machine: $PM_{init}$ = $N^r$.GetOne();
   The initial status of mapping: S = false;
2  foreach $VNF_p$ in $N^s$ do
3      Set the status of mapping to false, S = false;
4      $G_{pm} = PM_{init}$.GetGroup();
5      foreach $PM_i$ in $G_{pm}$ do
6          if $PM_i$ can hold $VNF_p$ then
7              Add $PM_i$, $VNF_p$ and the number of $VNF_p$ to $MS_{res}$;
8              Select the PLs between $PM_i$ and the previous $PM_{i-1}$ and add them into $MS_{res}$;
9              Set the status of mapping to true: S = true;
10             break;
11         end
12     end
13     if S is false then
14         $MS_{res} = \varnothing$;
15         break;
16     end
17 end
18 return $MS_{res}$;
```

Function 1: *GetGroup()* Function

```
Input: The physical machine: $PM_i$;
Output: The physical machine affinity group: $G_{pm}$;
1  Initialize: Get all PMs from $PN^r$($N^r$, $L^r$, $S^r$): $N^r$;
2  foreach $PM_j$ in $N^r$ do
3      if $PM_j$ is empty then
4          if $\lfloor\frac{2(i-1)}{k}\rfloor = \lfloor\frac{2(j-1)}{k}\rfloor$ then
5              Add $PM_j$ to affinity group $G_a$;
6          end
7          else if $\lfloor\frac{4(i-1)}{k^2}\rfloor = \lfloor\frac{4(j-1)}{k^2}\rfloor$ then
8              Add $PM_j$ to affinity group $G_b$;
9          end
10         else if $\lfloor\frac{4(i-1)}{k^2}\rfloor \neq \lfloor\frac{4(j-1)}{k^2}\rfloor$ then
11             Add $PM_j$ to affinity group $G_c$;
12         end
13         else
14             Add $PM_j$ to affinity group $G_d$;
15         end
16     end
17     else
18         Add $PM_j$ to affinity group $G_d$;
19     end
20 end
21 Add $G_a$, $G_b$, $G_c$ and $G_d$ to $G_{pm}$ in order;
22 return $G_{pm}$;
```

physical network. However, it is a difficult problem on how to find another suitable PM when the previous PM cannot hold the VNF. In this stage, we map VNFs based on affinity groups, thereby reducing PL cost.

Secondly, in the backup stage, we evaluate the availability of SFC and provide backup for VNFs. We use the proposed availability model to evaluate the availability of SFC. Then, we provide backup for VNFs until the availability of SFC can meet the availability requirements. Therefore, the backup model is a key technology in this stage. The traditional backup model simply provides backup for VNF and path resulting in high resource consumption. In this stage, we propose the JPV backup model in which a component can provide backup for two components to reduce PM cost and PL cost.

B. Mapping Stage

In Algorithm 1, we describe the mapping stage, which is responsible for mapping SFC to physical data center network.

The input of this algorithm is SFCR and physical network. We define the mapping result as a triple: *MS_{res}(PM_{res}, VNF_{res}, PL_{res})*. The PM_{res} is the set of PMs, and the VNF_{res} consists of VNF and the number of redundant VNFs. The PL_{res} consists of PLs between PM and the previous PM. For example, two VNF1 and one VNF2 are mapped to PM1 while one VNF3 is mapped to PM2. The PLs between PM1 and PM2 are 1-29-2. The MS_{res} adds two results: *(PM1, ((VNF1, 2), (VNF2, 1)), ())* and *(PM2, ((VNF3, 1)), (1-29-2))*.

In the initial state, we randomly get a PM from N^r as PM_{init}. At line 2 in Algorithm 1, VNFs in SFCR can be mapped to PMs. For each VNF, a PM group G_{pm} is provided for mapping (line 4 in Algorithm 1). The G_{pm} is generated according to the affinity model (mentioned in Section III-E), which will be described in detail in Function 1. At line 5 in Algorithm 1, the algorithm traverses the entire G_{pm} until there is a PM that can be mapped. If the VNF can be successfully mapped, we add the mapping information into MS_{res} (line 7-8 in Algorithm 1). After these operations, the traversal will be exited (line 10 in Algorithm 1) and the mapping of the next VNF will start.

However, the task will fail as long as one of the VNFs cannot be successfully mapped to PM. Therefore, we define the mapping status *S* to indicate whether the VNF is successfully mapped. At line 3 in Algorithm 1, we set *S* to false before mapping a VNF. If this VNF is successfully mapped, we set *S* to true. Then at lines 13-16 in Algorithm 1, *S* will be judged before mapping of next VNF. If *S* is false, the task fails to exit.

The details on how to generate the G_{pm} are shown in *GetGroup()* Function. This function is designed to generate the affinity groups of a PM, so the input of this function is a specific PM and the output is G_{pm}.

In this function, we divide all PMs into four groups. Groups A, B, and C are divided according to the affinity model in Section III-E (line 4-12 in Function 1). Group D contains the remaining PMs, including PMs that already contain VNF and the others (line 13-19 in Function 1). At line 21 in Function 1, the four groups A, B, C, and D are added into G_{pm} in order.

Algorithm 2: Backup Stage

```
Input: The SFC request s: $\mathbb{SR}^s$ = ($N^s$, $L^s$, $A^s$);
       The physical network r: $\mathbb{PN}^r$ = ($N^r$, $L^r$, $S^r$);
Output: The backup result: $BS_{res}$;
1  Initialize: The initial backup data: $BS_{res} = MS_{res}$;
2  $\begin{cases} A^s_{ci} = A^s_{n_i} A^r_{v_p} \\ A_{sfc} = \prod A^s_{c_i} \end{cases}$
3  if $A_{sfc} \geq A^s$ then
4      return $BS_{res}$;
5  end
6  else
7      foreach $VNF_p$ and $VNF_q$ in $N^s$ do
8          Get the PM that $VNF_p$ runs on, indicated as $PM_p$;
9          $G_{pm} = PM_p.GetGroup()$;
10         foreach $PM_b$ in $G_{pm}$ do
11             if $PM_b$ can hold $VNF_p$ and $VNF_q$ then
12                 Add $PM_b$, $VNF_p$, $VNF_p$, number of
                   $VNF_p$ and $VNF_q$ to $BS_{res}$;
13                 Select the PLs between $PM_b$ and the
                   previous $PM_{b-1}$, and the PLs between
                   $PM_b$ and the next $PM_{b+1}$. Then, add
                   them into $BS_{res}$;
14                 break;
15             end
16         end
17         $\begin{cases} A_b = A_{n_b} A_{v_p} A_{v_q} \\ A_i = A_{n_i}(1-(1-A_{v_p})^{\theta_i}) \\ A_j = A_{n_j}(1-(1-A_{v_q})^{\theta_j}) \\ A_c = 1-(1-A_b)(1-A_i A_j) \\ A_{sfc} = \prod A^s_{c_i} \end{cases}$
18         if $A_{sfc} \geq A^s$ then
19             return $BS_{res}$;
20         end
21     end
22 end
23 if $A_{sfc} \geq A^s$ then
24     return $BS_{res}$;
25 end
26 else
27     Return $A_{sfc}$ and ask whether $A^s$ can be reduced;
28 end
```

In Function 1, we propose the concept of the affinity group. In the mapping stage, the numbers of PLs used between two PMs can be reduced effectively by using affinity groups.

C. Backup Stage

In Algorithm 2, we describe the backup stage, which is designed to meet the availability requirements of SFC.

At the backup stage, we use the same triple as the mapping stage, indicated as BS_{res}, to record the backup results. In the initial state, we assign the MS_{res} in the mapping stage to BS_{res} (line 1 in Algorithm 2).

Then, we evaluate the availability of SFC by using Eqs. (2) and (12) (line 2 in Algorithm 2). If the availability of SFC is not less than the availability requirements, the SFC meets the requirements and backup stage ends (line 3-5 in Algorithm 2). However, the availability of SFC often does not meet the requirements, so the backup process is required (line 6-22 in Algorithm 2).

During the backup process, we use the JPV backup model presented in Section III-D. At line 7 in Algorithm 2, we provide backup for two VNFs each time, repeating the above operations until the availability meets the requirements. Similarly, we first get the PM running VNF and then use Function 1 to generate the affinity groups of this PM (line 8-9 in Algorithm 2). At line 11 in Algorithm 2, the backup VNFs that need to be provided are mapped according to the affinity groups. If the VNFs can be successfully mapped, we add the backup information (such as PM, VNFs, the number of VNFs and the PLs used) into the BS_{res} (line 12-13 in Algorithm 2). Then, the backup of two VNFs is finished (line 14 in Algorithm 2).

After providing backup, we need to update the availability of SFC according to Eq. (9) in Section IV-C (line 17 in Algorithm 2). If the availability meets the requirements, the algorithm exits successfully (line 18-20 in Algorithm 2). Otherwise, the backup operation of the next two VNFs starts.

However, the availability of SFC may still not meet availability requirements even though we provide all backup (lines 26-28 in Algorithm 2). At line 27 in Algorithm 2, the algorithm returns the availability of SFC and asks whether the availability requirements can be reduced.

During the backup stage, we propose the JPV backup model to increase availability with less resource. Similarly, we map the backup VNFs based on affinity groups, reducing the number of PLs used.

D. Complexity Analysis

In this subsection, we give a detailed time complexity analysis of ABA. We introduce two variables, δ and η, which are the number of mapped VNFs and the number of PMs.

Since both Algorithms 1 and 2 call Function 1, we first consider the time complexity of Function 1. For Function 1, the *foreach* (line 2 in Function 1) runs η times, and line 21 in Function 1 also runs η times. Thus, the total time complexity of Function 1 is:

$$\eta + \eta = 2\eta \tag{20}$$

We can observe that the time complexity of Function 1 is at the level of O(η).

In Algorithm 1, the *foreach* (line 2 in Algorithm 1) runs δ times. Then the Algorithm 1 calls the *GetGroup()* Function (line 4 in Algorithm 1). At line 5 in Algorithm 1, another *foreach* runs η times. Therefore, the time complexity of Algorithm 1 is:

$$\delta \cdot (2\eta + \eta) = 3\delta\eta \tag{21}$$

We can conclude that the time complexity of Algorithm 1 is at the level of O($\delta\eta$).

In Algorithm 2, the calculation of availability runs δ times (line 2 in Algorithm 2). At line 7 in Algorithm 2, each time it selects two VNFs, so it runs $\delta/2$ times. Then the algorithm calls the *GetGroup()* Function (line 9 in Algorithm 2). At line 10 in Algorithm 2, the *foreach* runs η times. Then update the availability of SFC, which runs $\delta/2$ times (line 17 in Algorithm 2). Therefore, the total time complexity of Algorithm 2 is:

$$\delta + \frac{\delta}{2} \cdot \left(2\eta + \eta + \frac{\delta}{2}\right) = \delta + \frac{3}{2}\delta\eta + \frac{\delta^2}{4} \tag{22}$$

It can be seen that the time complexity of Algorithm 2 is at the level of O($\delta(\delta + \eta)$).

In general, Eqs. (21) and (22) indicate the time complexity of Algorithms 1 and 2. Therefore, the total time complexity of ABA is at the level of O($\delta(\delta + \eta)$).

VI. Performance Evaluation

In this section, we evaluate the performance of our proposed JPV backup model and ABA in detail.

A. Comparison of Backup Models and Placement Algorithms

1) Backup Models: We compare JPV backup model with VNF, Path, PV and JP backup models (described in Section III-D).

- VNF backup model is designed to provide backup VNFs for working VNFs.
- Path backup model is responsible for providing backup paths for working paths.
- PV (Path-VNF) [21] backup model is a simple combination of Path backup model and VNF backup model.
- JP (Joint Protection) [14] backup model combines path backup and VNF backup. It provides backup for two non-adjacent PMs.
- JPV (Joint Path-VNF) backup model also combines path backup and VNF backup. Different from JP backup model, JPV backup model provides backup for two adjacent PMs.

2) Placement Algorithms: We compare ABA with Heuristic and LVSCP. And we keep RBA as the baseline.

- Heuristic [15] formulates the SFC mapping problem as an ILP (Integer Linear Programming) model. And Heuristic provides an optimal iterative heuristic solution to map each SFC request. This algorithm calculates the optimal placement position before mapping each VNF. Therefore, the placement of each VNF in the current network state is the optimal solution.
- LVSCP (Local VNF Service Chain Placement) [18] aims to keep all the VNFs as close as possible with redundancy and location constraints.
- RBA (Random-Based Algorithm) maps each VNF randomly.
- ABA (Affinity-Based Algorithm) maps each VNF based on the affinity model.

We run each of the four algorithms with backup model for 100 times. All of the data are the average results and almost unaffected by accidental events.

B. Evaluation Setup

We evaluate the backup models and placement algorithms using a laptop of windows 10 with 2.2 GHz Intel Core i5 processor and 8GB memory. We implement the models and algorithms in Java based on Alevin [13], a widely used environment for NFV resource allocation. The Alevin uses Barabasi-Albert generator to generate random network topologies. In this paper, we develop a generator that generates fat-tree network topologies based on Alevin. Here are the parameters of the models and algorithms.

1) SFC Request: In this paper, we map SFCR one by one. After each SFCR is deployed, the state of the network changes. Therefore, we update the current network state before mapping each SFCR. Each SFCR consists of two to six VNFs. The flow in SFCR is controlled by SDN controller [30]. Each VNF can provide one network function and requires three types of physical resource. There are eight types of VNFs with availability between [0.99, 0.9999]. The logical link has a bandwidth demand between [10, 30]. And the computing and storage requirements are between [10, 20] and [5, 10], respectively. By referring to Google Apps [31] and other literature [16], [21], we divide SFC availability requirements into four levels among $\{0.99, 0.999, 0.9999, 0.99999\}$.

2) Fat-Tree Topology: As we described, all of the four typical types of data center architectures can be divided into multiple domains. Therefore, in this paper, we use a 3-layer fat-tree topology as an example [28]. As we discussed in Section III, the size of fat-tree is determined by the number of ports in the switches. In this evaluation, we set the default value of k to 8. Therefore, in an 8-ray fat-tree topology, there are 16 core switches, 128 PMs and 8 pods, with 4 aggregation switches, 4 edge switches in each pod.

3) Physical Device: Each PM contains three types of resource including computing, memory, and bandwidth, with the capacity between [80, 100]. In Section III, we propose an availability model that considers both physical device failures and VNF failures. In this evaluation, we refer to the data of Cisco switches and Intel servers to determine the availability of physical device, as shown in Table I. Thus, the availability of switches is distributed within [0.9999, 0.99999], and the availability of PMs is distributed within [0.999, 0.99999].

4) Network Traffic: In the data center network, traffic changes over time. In [32], the authors develop four programs to model the gathered data and generate the required data. By referring to this well-known data-generation model. In this paper, we use traffic matrix tracking to generate time-varying traffic for the data center network [33].

As network traffic can change dynamically over time, scaling in/out becomes a challenging problem remaining to be solved [34]. Most of the existing work formulates the problem as ILP (Integer Linear Programming). However, ILP requires a long time to find an optimal solution. Therefore, in this paper, we are going to accommodate scaling in/out requirements by using a genetic algorithm [35].

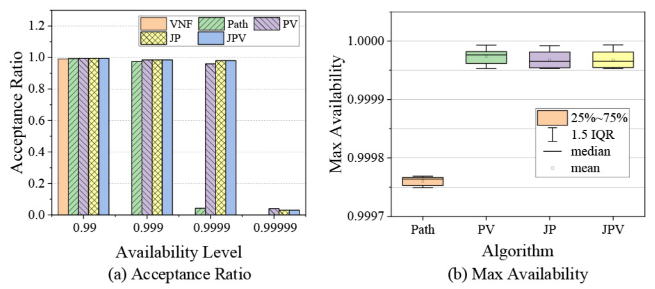


Fig. 5. Performance improvement: (a) Acceptance ratio; (b) Max availability.

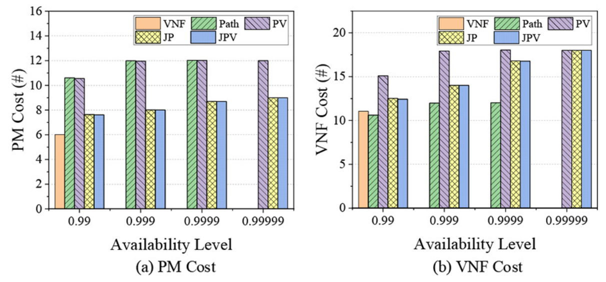


Fig. 6. Resource consumption: (a) PM cost; (b) VNF cost.

C. Availability Guarantee

In this subsection, we focus on the availability guarantee. We evaluate the JPV backup model and compare it with VNF, Path, PV and JP backup models. First, we evaluate the performance improvement of backup models, such as acceptance ratio and maximum availability that can be achieved. Then we compare the resource consumption of these backup models, including the number of VNFs and PMs used.

1) Performance Improvement: As described in Section III, JP and JPV are the same in terms of performance improvement including acceptance ratio and maximum availability.

We evaluate the acceptance ratio of five backup models under ABA, as shown in Fig. 5(a). When the availability requirements are two 9s, all of the backup models are close to 1. When three 9s, the four backup models are close to 1 while VNF backup model is 0. It means that VNF backup model does not meet three 9s. Similarly, Path backup model does not satisfy four 9s. For the five 9s, the acceptance ratios of JP and JPV are 0.03, while PV is 0.04. In this case, we can think that PV, JP, and JPV cannot meet five 9s. In summary, JPV and PV backup models perform the same and outperform than VNF and Path backup models in terms of acceptance ratio.

As Fig. 5(b) shows, we are concerned with the average results. The maximum availability of VNF backup model is 0.9967, which satisfies two 9s. For ease of observation, data of VNF backup model does not appear in Fig. 5(b). The Path backup model is 0.99976, which satisfies three 9s. The availability of PV is 0.999973 while JP and JPV are 0.999967. We can conclude that PV, JP, and JPV can meet four 9s. It shows that the impact of Δ (mentioned in Section III) between PV and JPV on availability can be ignored.

In summary, our proposed JPV backup model is superior to other backup models in terms of performance improvement (i.e., high acceptance ratio and high availability).

2) Resource Consumption: We evaluate resource consumption including PM and VNF cost. As we described in Section III-D, the extended JP and JPV are the same in terms of PM cost and VNF cost.

The PM cost means the number of PMs used, as shown in Fig. 6(a). We can observe that the PM costs of Path and PV are close, and they are higher than JPV under the same availability requirements. This proves that a backup model that one component provides backup for two components can efficiently reduce PM cost. As the availability requirements increase, the PM cost of the backup model increases, but it is not obvious. It shows that all backup models require more PMs to achieve higher availability.

Fig. 6(b) shows the number of VNFs used in the five backup models. We can observe that the VNF cost of PV and JPV is higher than VNF and Path. It means that PV and JPV backup models use more redundant VNFs to achieve higher availability. However, compared with PV backup model, the VNF cost of JPV is significantly lower in the case of two 9s to four 9s. It proves that our proposed JPV backup model is superior to PV backup models in VNF cost, as we mentioned in Section III. However, in the case of five nines, the VNF costs of PV and JPV are close. It means that in the case of five 9s, both PV and JPV provide all redundant VNFs. As we mentioned in Section III, the VNF costs of PV and JPV are the same when all redundant VNFs are provided.

To improve availability, we prefer to use PV, JP, and JPV backup models. As Figs. 6(a) and 6(b) show, JP and JPV backup models perform the same and outperform PV backup model in terms of PM cost and PL cost.

D. PL Cost

In this subsection, we first analyze the impact of backup model and mapping algorithm on the PL cost separately. Then we comprehensively analyze the performance of JPV backup model with ABA.

1) Backup Model: In this subsection, we evaluate the PL costs of five backup models under RBA (Random-Based Algorithm) and ABA (Affinity-Based Algorithm).

Fig. 7(a) shows the PL costs of five backup models under RBA. First, we can observe that the PL cost of each backup model under different availability requirements has remained essentially unchanged. Then, we can conclude that the VNF backup model has the lowest PL cost because it does not use additional PLs. The Path, PV and JP backup models have the same PL cost and are the highest. For our proposed JPV backup model, the PL cost is significantly lower than Path, PV and JP backup models. It is consistent with our discussion about resource consumption (Table II) in Section III.

As Fig. 7(b) shows, we evaluate the PL costs of five backup models under ABA. For the VNF backup model, there is no PL cost during the backup stage, so the PL cost is the same under different availability requirements. However, in the other four backup models, the PL costs tend to increase as availability requirements increase. It means that the backup model needs to use more resource when the availability requirements are higher. For the PL cost, VNF is the lowest, and Path and PV

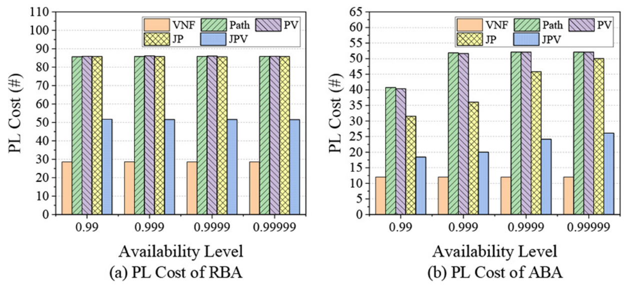


Fig. 7. PL cost: (a) PL cost in RBA; (b) PL cost in ABA.

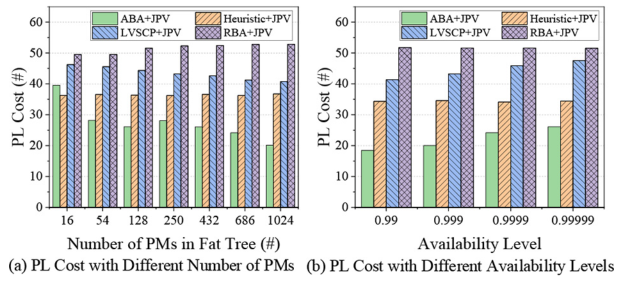


Fig. 8. PL cost: (a) Different numbers of PMs; (b) Different availability levels.

are the same and highest. JP reduces the PL cost compared with Path and PV. However, in ABA, we cannot attribute all optimization of the PL cost to backup models.

In general, we can conclude from Figs. 7(a) and 7(b) that both backup model and mapping algorithm have an impact on PL cost. Our proposed JPV backup model is significantly better than JP backup model since it does not reduce the PL cost in RBA. The optimized performance of JP under ABA is only half of that of JPV. Fig. 7(a) shows the PL cost in RBA, thus removing the impact of ABA on PL cost. From Fig. 7(a), we can confirm the superiority of providing backup for two consecutive components in JPV. We can conclude that JPV backup model can reduce the PL cost by 40% compared with Path and PV backup models.

2) Mapping Algorithm: In this subsection, we discuss the impact of the mapping algorithm on PL cost. We evaluate the performance of the mapping algorithm for different numbers of PMs and availability requirements.

Fig. 8(a) shows the PL costs in RBA and Heuristic do not change substantially at different numbers of PMs. However, the PL costs in ABA and LVSCP are sensitive to the number of PMs. The more PMs in the topology, the lower PL cost. It means that the more PMs, the more PMs in each affinity group. This makes it easier to find PMs in the same affinity group, so the PL cost is reduced. This also explains why the PL cost of ABA is significantly higher than the others when the number of PM is 16. We can also observe that the order of effects of reducing the PL costs is ABA, Heuristic, LVSCP, and RBA. It is worth mentioning that although the Heuristic is an optimal iterative algorithm. However, it can get the optimal solution of each VNF in the current network state rather than the overall most optimal solution.

Similarly, the PL costs of RBA and Heuristic are not sensitive to the availability levels, as shown in Fig. 8(b). In ABA and LVSCP, the PL costs increase as availability requirements increase. It means that the backup model uses more resource to guarantee higher availability. Therefore, the PL cost increases as the availability requirements increase. From Fig. 8(b), we can observe that the lower the availability requirements, the more obvious impact of ABA on reducing the PL cost. We can conclude that ABA has the best effect on reducing the PL cost, followed by Heuristic, then LVSCP, and finally RBA.

In summary, we can conclude that both the number of PMs and availability requirements have an impact on ABA and LVSCP. Figs. 8(a) and 8(b) show that our proposed ABA solution performs better than Heuristic, LVSCP, and RBA in reducing the PL cost.

3) Backup Model and Mapping Algorithm: In this subsection, we evaluate the impact of both backup model and mapping algorithm on the PL cost. We evaluate the PL costs of the five backup models under ABA and RBA with different availability requirements.

As Fig. 9(a) shows, VNF backup model only has data under two 9s requirements, which means that VNF backup model only satisfies two 9s. Similarly, we can conclude that JPV backup model is superior to VNF, Path, and PV backup models in terms of maximum availability. As Figs. 9(a) to 9(e) show, the PL cost of VNF backup model is the lowest compared with the other four backup models. In addition to VNF backup model, the PL cost of JPV model is lower than the other three backup models, whether under ABA or RBA. It demonstrates that compared with Path, PV and JP backup models, our proposed JPV backup model performs best in optimizing the PL cost.

As shown in Figs. 9(a) to 9(e), under RBA, the curves under different availability requirements overlap. It means that although the availability requirements are different, the PL cost of each backup model is unchanged. The PL cost increases slowly as the number of PMs increases. However, under ABA, when the availability requirements are different, the PL cost of each algorithm is different. The PL cost decreases as the number of PMs increases. In each backup model, the PL cost of ABA is significantly lower than RBA.

In summary, as Figs. 9(a) to 9(e) show, ABA can effectively reduce the PL cost. Both availability requirements and the number of PMs have an impact on the performance of ABA. We can also conclude that there is a trade-off between availability guarantee and resource optimization. Therefore, we propose ABA to reduce the PL cost. Compared with RBA, ABA can effectively reduce the PL cost. In order to improve availability, we can use a more efficient backup model. For PL cost, we can use a better mapping algorithm and reduce the availability requirements.

4) Execution Time: Fig. 9(f) shows the execution time of ABA and RBA with different numbers of PMs. We can observe that ABA runs faster than RBA at the same number of PMs, although it is not obvious. As the number of PMs increases, the execution times of the two algorithms also increase. It means that as the number of PMs increases, the longer it takes to find a PM that can be mapped.

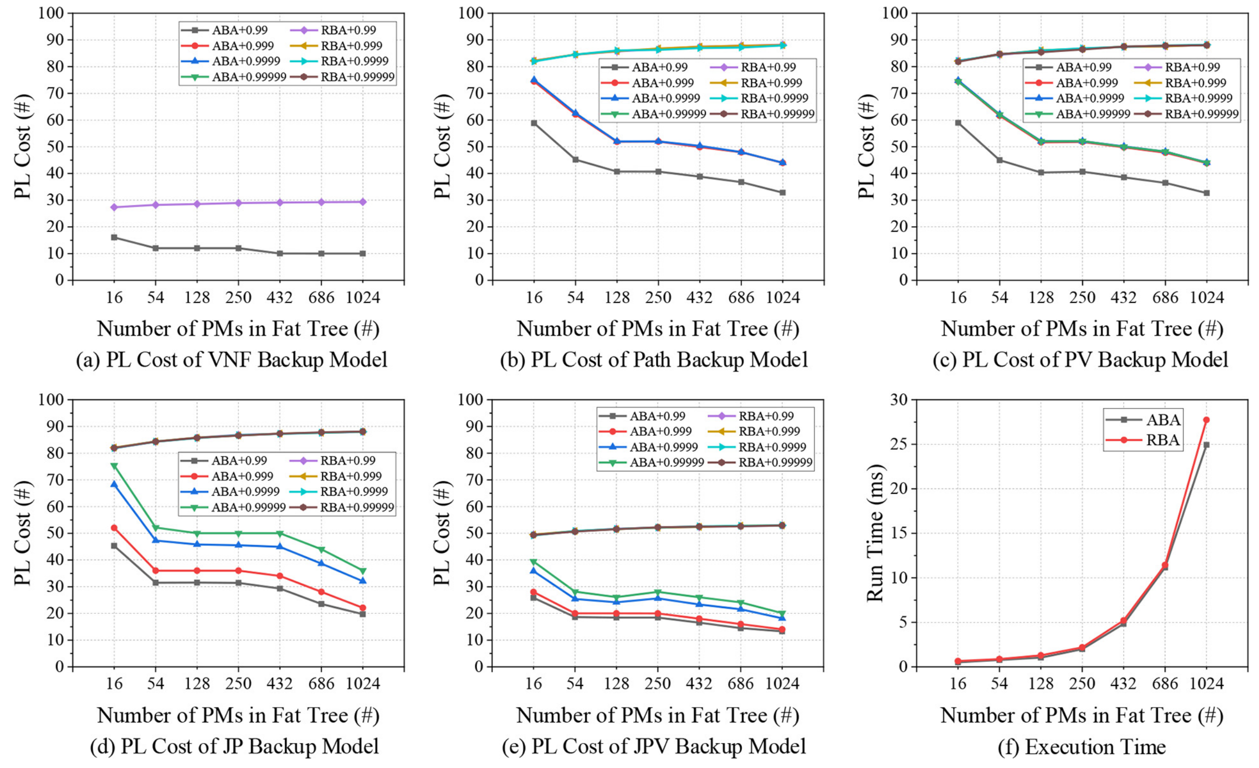


Fig. 9. PL cost: (a) VNF; (b) Path; (c) PV (d) JP; (e) JPV. (f) Execution time of ABA and RBA.

VII. Conclusion

This paper addresses the VNF placement problem in data center networks considering availability guarantee and resource optimization. We define an availability model that takes both physical device failures and VNF failures into consideration. We propose a JPV backup model to improve the availability of SFC. And we design ABA to reduce resource consumption. The evaluation results show that our proposed JPV backup model can achieve higher availability with less resource. And ABA can efficiently reduce PL cost.

However, there are still some limitations in our work. Firstly, we evaluate our solutions in the simulation environment. We do not implement the backup models and placement algorithms in a real NFV platform. Secondly, there are more requirements and constraints needed to be involved. Consequently, our future work will focus on the above aspects.

References

[1] *Network Functions Virtualisation (NFV)*. Accessed: Oct. 2014. [Online]. Available: https://portal.etsi.org/Portals/0/TBpages/NFV/Docs/NFV_White_Paper3.pdf

[2] H. Moens and F. D. Turck, "Customizable function chains: Managing service chain variability in hybrid NFV networks," *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 4, pp. 711–724, Dec. 2016.

[3] J. Fan *et al.*, "A framework for provisioning availability of NFV in data center networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 10, pp. 2246–2259, Oct. 2018.

[4] D. Wu, Y. Xia, X. Sun, X. S. Huang, S. Dzinamarira, and T. S. E. Ng, "Masking failures from application performance in data center networks with shareable backup," in *Proc. Conf. ACM Special Interest Group Data Commun. (SIGCOMM)*, 2018, pp. 176–190.

[5] P. Gill, N. Jain, and N. Nagappan, "Understanding network failures in data centers: Measurement, analysis, and implications," in *Proc. Conf. ACM Special Interest Group Data Commun. (SIGCOMM)*, 2011, pp. 350–361.

[6] Z. Guo, W. Feng, S. Liu, W. Jiang, Y. Xu, and Z.-L. Zhang, "RetroFlow: Maintaining control resiliency and flow programmability for software-defined WANs," in *Proc. Int. Symp. Qual. Service (IWQoS)*, 2019, pp. 1–10.

[7] M. Otokura, K. Leibnitz, Y. Koizumi, D. Kominami, T. Shimokawa, and M. Murata, "Evolvable virtual network function placement method: Mechanism and performance evaluation," *IEEE Trans. Netw. Service Manag.*, vol. 16, no. 1, pp. 27–40, Mar. 2019.

[8] M. Mechtri, C. Ghribi, and D. Zeghlache, "A scalable algorithm for the placement of service function chains," *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 3, pp. 533–546, Sep. 2016.

[9] *High Availability for OPNFV*. Accessed: Mar. 2019. [Online]. Available: https://wiki.opnfv.org/display/availability

[10] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, "Network function virtualization: Challenges and opportunities for innovations," *IEEE Commun. Mag.*, vol. 53, no. 2, pp. 90–97, Feb. 2015.

[11] *ETSI GS NFV-REL 001 V1.1.1*. Accessed: Jan. 2015. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/NFV-REL/001_099/001/01.01.01_60/gs_NFV-REL001v010101p.pdf

[12] W. Rankothge, F. Le, A. Russo, and J. Lobo, "Optimizing resource allocation for virtualized network functions in a cloud center using genetic algorithms," *IEEE Trans. Netw. Service Manag.*, vol. 14, no. 2, pp. 343–356, Jun. 2017.

[13] J. Gil-Herrera and J. F. Botero, "Resource allocation in NFV: A comprehensive survey," *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 3, pp. 518–532, Sep. 2016.

[14] J. Fan, Z. Ye, C. Guan, X. Gao, K. Ren, and C. Qiao, "GREP: Guaranteeing reliability with enhanced protection in NFV," in *Proc. ACM SIGCOMM Workshop Hot Topics Middleboxes Netw. Funct. Virtualization*, 2015, pp. 13–18.

[15] J. Fan, M. Jiang, and C. Qiao, "Carrier-grade availability-aware mapping of service function chains with on-site backups," in *Proc. IEEE/ACM 25th Int. Symp. Qual. Service (IWQoS)*, Vilanova i la Geltrú, Spain, 2017, pp. 1–10.

[16] J. Fan, C. Guan, Y. Zhao, and C. Qiao, "Availability-aware mapping of service function chains," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Atlanta, GA, USA, 2017, pp. 1–9.

[17] L. Qu, C. M. Assi, K. B. Shaban, and M. J. Khabbaz, "A reliability-aware network service chain provisioning with delay guarantees in NFV-enabled enterprise datacenter networks," *IEEE Trans. Netw. Service Manag.*, vol. 14, no. 3, pp. 554–568, Sep. 2017.

[18] S. Herker, X. An, W. Kiess, S. Beker, and A. Kirstädter, "Data-center architecture impacts on virtualized network functions service chain embedding with high availability requirements," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, San Diego, CA, USA, 2015, pp. 1–7.

[19] J. Sun *et al.*, "A reliability-aware approach for resource efficient virtual network function deployment," *IEEE Access*, vol. 6, pp. 18238–18250, 2018.

[20] J. Kang, O. Simeone, and J. Kang, "On the trade-off between computational load and reliability for network function virtualization," *IEEE Commun. Lett.*, vol. 21, no. 8, pp. 1767–1770, Aug. 2017.

[21] J. Kong *et al.*, "Guaranteed-availability network function virtualization with network protection and VNF replication," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Singapore, 2017, pp. 1–6.

[22] M. C. Luizelli, L. R. Bays, L. S. Buriol, M. P. Barcellos, and L. P. Gaspary, "Piecing together the NFV provisioning puzzle: Efficient placement and chaining of virtual network functions," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manag. (IM)*, Ottawa, ON, Canada, 2015, pp. 98–106.

[23] W. Rankothge, J. Ma, F. Le, A. Russo, and J. Lobo, "Towards making network function virtualization a cloud computing service," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manag. (IM)*, Ottawa, ON, Canada, 2015, pp. 89–97.

[24] Z. Ye, X. Cao, J. Wang, H.-F. Yu, and C. Qiao, "Joint topology design and mapping of service function chains for efficient, scalable, and reliable network functions virtualization," *IEEE Netw.*, vol. 30, no. 3, pp. 81–87, May/Jun. 2016.

[25] X. Meng, V. Pappas, and L. Zhang, "Improving the scalability of data center networks with traffic-aware virtual machine placement," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, San Diego, CA, USA, 2010, pp. 1154–1162.

[26] R. Potharaju and N. Jain, "When the network crumbles: An empirical study of cloud network failures and their impact on services," in *Proc. 4th Annu. Symp. Cloud Comput. (SOCC)*, 2013, pp. 1–17.

[27] Z. Guo *et al.*, "Balancing flow table occupancy and link utilization in software-defined networks," *Future Gener. Comput. Syst.*, vol. 89, pp. 213–223, Dec. 2018.

[28] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 63–74, 2008.

[29] N. Bouten, R. Mijumbi, J. Serrat, J. Famaey, S. Latré, and F. D. Turck, "Semantically enhanced mapping algorithm for affinity-constrained service function chain requests," *IEEE Trans. Netw. Service Manag.*, vol. 14, no. 2, pp. 317–331, Jun. 2017.

[30] Z. Guo, W. Chen, Y.-F. Liu, Y. Xu, and Z.-L. Zhang, "Joint switch upgrade and controller deployment in hybrid software-defined networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1012–1028, May 2019.

[31] *Google Apps Service Level Agreement*. Accessed: Aug. 2019. [Online]. Available: http://www.google.com/apps/intl/en/terms/sla.html

[32] W. Rankothge, F. Le, A. Russo, and J. Lobo, "Data modelling for the evaluation of virtualized network functions resource allocation algorithms," 2017. [Online]. Available: arXiv:1702.00369.

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

[34] J. Ma, W. Rankothge, C. Makaya, M. Morales, F. Le, and J. Lobo, "An overview of a load balancer architecture for VNF chains horizontal scaling," in *Proc. 14th Int. Conf. Netw. Service Manag. (CNSM)*, Rome, Italy, 2018, pp. 323–327.

[35] J. Lobo, W. Rankothge, and H. Ramalhinho, "On the scaling of virtualized network functions," in *Proc. IFIP/IEEE Symp. Integr. Netw. Service Manag. (IM)*, Arlington, VA, USA, 2019, pp. 125–133.

Meng Wang (Student Member, IEEE) received the B.S. degree from Beijing Jiaotong University, Beijing, China, in 2016. He is currently pursuing the Ph.D. degree with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing. His current research interests include network function virtualization, network slicing, and resource allocation managements.

Bo Cheng (Member, IEEE) received the Ph.D. degree in computer science from the University of Electronic Science and Technology of China in 2006. He is currently a Professor with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. His research interests include multimedia communications and services computing.

Junliang Chen is a Professor with the Beijing University of Posts and Telecommunications. His research interest are in the area of service creation technology. He was elected as a member of the Chinese Academy of Science in 1991 and the Chinese Academy of Engineering in 1994.