Availability-aware Service Chain Composition and Mapping in NFV-enabled Networks

Meng Wang, Bo Cheng, Shuai Zhao, Biyi Li, Wendi Feng, Junliang Chen
State Key Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications
Beijing, P.R. China
{*mengwang, chengbo, zhaoshuaiby, lbyi0402, logan, chjl*}*@bupt.edu.cn*

***Abstract*—Network Function Virtualization (NFV) is an emerging technology decouples network functions from hardware. Network service in NFV is deployed as a service chain, also known as Service Function Chain (SFC). SFC consists of an ordered set of Virtual Network Functions (VNFs). However, VNFs bring new challenges in providing network services with availability guarantee. In addition, in a customizable and dynamic NFV-enabled network, the composition and mapping of service chain are different from that of a traditional network. In this paper, we define an availability model that takes both hardware and VNF failures into consideration. Then we propose a Joint Path-VNF backup model to combine path and VNF backup in a joint way. And a priority-based algorithm is designed for service chain composition and mapping. Simulation results show that our proposed solutions can reduce resource consumption while guaranteeing availability.**

***Keywords*-Network Function Virtualization; Service Function Chain; Availability; Composition; Resource Allocation;**

I. Introduction

With the development of network services, there is a key challenge about how to deploy network services in an efficient way. The tightly coupled deployment of network functions on physical resources results in operators being bound by specific hardware. Network Function Virtualization (NFV) [1] is an emerging technology decouples network functions from hardware. By allowing various network functions to be virtualized, NFV decreases the Capital Expenditure (CAPEX) and Operating Expense (OPEX) significantly [2]. Based on the requirements of applications, network service is described as a service chain. The service chain, also known as Service Function Chain (SFC), consists of an ordered set of Virtual Network Functions (VNFs).

For SFC, it runs on physical nodes and contains several VNFs. Therefore, both hardware and VNF failures can result in the unavailability of SFC. For example, we assume that a linear service chain consists of 4 VNFs, each with the availability of 0.95. However, the availability of SFC is 0.95^4 = 0.81, which does not meet the availability requirements of most network services. To mask failure, redundancy is an effective technique [3]. In this paper, we use the active/standby redundancy model [4], in which a standby entity can be used if a VNF fails. And we can also provide backup for the working path, which is to add a backup path containing physical links and physical nodes. After providing backup for path and VNF, we deployed SFC to the physical network. There are requirements and dependencies between VNFs in SFC, which complicate the composition and mapping of SFC.

There are different kinds of backup models in the existing works to guarantee the availability of SFC [5], [6], [7], [8]. Moreover, multiple algorithms are also designed to optimize the composition and mapping of SFC [9], [10], [11], [12]. However, there are still some problems remaining to be solved. Firstly, the failures of the physical node are ignored in previous studies. We have to consider the failures of both hardware and VNF [13]. Secondly, path and VNF backup often appear separately. When the one achieves the highest performance, the other will become a bottleneck for performance improvement. Thirdly, for dynamic and customizable SFCs, traditional composition and mapping algorithms do not perform well.

Given these facts, we focus on the composition and mapping of SFC. We define an availability model and propose a novel backup model to guarantee the availability of SFC. Then, we design a priority-based algorithm to optimize the composition and mapping of SFC. In summary, the main contributions are as follows:

- We define an availability model to calculate the availability of SFC. It cannot be ignored that hardware failures may affect the availability of several VNFs in one SFC. Therefore, different from some existing works, we take both hardware and VNF failures into consideration. Although it makes our problem harder, it is closer to reality.
- We propose a Joint Path-VNF (JPV) backup model which jointly considers both path and VNF backup. In this model, one backup component can simultaneously provide backup for two components in SFC. By using JPV model, we can guarantee the availability of SFC with fewer resources such as VNFs, physical nodes and physical links.
- We design a priority-based algorithm to optimize the composition and mapping of SFC. In this algorithm,

978-1-7281-2717-0/19/$31.00 ©2019 IEEE
DOI 10.1109/ICWS.2019.00028

VNF dependency is converted to VNF priority. There might be several forwarding graphs generated according to the priorities between VNFs. Then, the SFC is mapped in the physical network based on an optimal forwarding graph.

The rest of this paper is organized as follows. Section II discusses the related works. Section III describes the system model. Section IV describes and formulates the problem. Section V describes the proposed priority-based algorithm. Section VI evaluates the performance of proposed solutions. Finally, Section VII concludes this study.

II. Related Works

In this paper, we focus on the composition and mapping of SFC considering availability guarantee and resource optimization. Therefore, we first discuss the existing works on the availability model and backup model. Then we describe the resource allocation in NFV, which involves the composition and mapping of SFC.

The first thing we need in availability guarantee is the availability model. In [7], [14], the authors define an optimal availability-aware SFC mapping problem. They present an algorithm that can minimize resource consumption while guaranteeing the availability of SFC. And they define an availability model that considers VNF failures. Herker *et. al.* [15] mention physical node failures in data center networks, but do not provide a clear availability model that considers both physical node and VNF failures. Sun *et. al.* [16] propose an algorithm to ensure reliability while taking both node and link failures into consideration.

Most of the existing works guarantee availability by proposing backup models and algorithms. Kang *et. al.* [17] study the tradeoff between computational due and reliability for VNF. In [18], the authors optimize traffic routing for VNF placement problem. Kong *et. al.* [8] propose a protection mechanism that combines both VNF replicas and backup path protection to guarantee the availability of SFC. They combine path and VNF backup for the first time, but it is just a simple combination of the two backup methods. In [6], the authors propose a joint protection model that uses one physical node to provide backup for two physical nodes. This model is similar to our JPV model, which uses one backup physical node. However, the joint protection model mainly focuses on path backup. Besides, the joint protection model requires more physical links.

For resource allocation in NFV, Herrera *et. al.* [19] define the NFV resource allocation problem as three phases, including VNF chain composition, VNF forwarding graph embedding and VNF scheduling. The VNF chain composition and VNF forwarding graph embedding correspond to the composition and mapping of SFC we are concerned with. In [10], the VNF is placed in a scalable and coordinated way. The authors summarize this problem and propose a mapping method in a random physical network.

The SFC needs to be mapped in the physical network, which is similar to the well-known VNE (Virtual Network Embedding) problem [20]. In traditional VNE problem, the input is a service chain that is static and predefined. However, the input of SFC mapping is a forwarding graph, which is defined according to the SFC request. Therefore, the task of SFC composition is to effectively generate a suitable forwarding graph that satisfies the requirements and dependencies between VNFs.

III. System Model

In this section, we describe the system model. Firstly, we introduce network model and SFC model. Then, we define an availability model. Finally, we describe four backup models including VNF, Path and PV (Path-VNF) and JPV.

A. Network Model

In this paper, we use $PN^r(N^r, L^r)$ to indicate the physical network. δ and ϵ indicate the number of physical nodes and physical links, respectively. N^r=$\{n_1^r, n_2^r \dots n_\delta^r\}$ indicates a set of physical nodes. And L^r=$\{l_1^r, l_2^r \dots l_\epsilon^r\}$ indicates the physical links. We use n_δ^r=$\{C_{n_\delta^r}^{cpu}, C_{n_\delta^r}^{mem}, a_\delta^r\}$ to indicate CPU capacity, memory capacity and the availability of physical node n_δ^r. One physical link connects two physical nodes. Therefore, we use l_ϵ^r=$\{n_{src}^r, n_{dest}^r, C_{l_\epsilon^r}^{bw}\}$ to indicate the source physical node, destination physical node and bandwidth capacity of physical link l_ϵ^r, respectively.

B. SFC Model

In this subsection, we use SFCR (SFC Request) to describe the attributes and requirements of SFC. $\mathbb{SR}$ indicates the set of SFCRs. One SFCR is indicated by $SR^s(V^s, L^s, A^s)$. A^s indicates the availability requirements of SFC including four levels: 0.99, 0.999, 0.9999, 0.99999 (Section VI). ζ and η indicate the number of VNFs and logical links. We use V^s=$\{v_1^s, v_2^s \dots v_\zeta^s\}$ to indicate the set of VNFs. L^s=$\{l_1^s, l_2^s \dots l_\eta^s\}$ indicates the logical links. Similarly, one logical link connects two VNFs. l_η^s=$\{v_{src}^s, v_{dest}^s, bw_{l_\eta^s}\}$ indicates the source VNF, destination VNF and bandwidth consumption of logical link l_η^s, respectively.

For VNF, v_ζ^s=$\{cpu_{v_\zeta^s}, mem_{v_\zeta^s}, a_\zeta^s, p_\zeta^s\}$ indicates the attributes and requirements of VNF v_ζ^s. We use $cpu_{v_\zeta^s}$ and $mem_{v_\zeta^s}$ to indicate CPU and memory consumption. a_ζ^s is the availability of VNF v_ζ^s. And p_ζ^s indicates the priority of this VNF, which is used to optimize the composition and mapping of SFC.

C. Availability Model

In this paper, we define an availability model that is based on elementary probability calculus. Besides, our proposed availability model takes both hardware and VNF failures into consideration.

First, we describe the availability of physical node and VNF, then define the availability of SFC.

1) Availability of Physical Node and VNF: In this model, the status of physical node and VNF 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) [21], respectively. Then the availability [22] of physical node and VNF can be characterized as follows:

$$A = \frac{Uptime}{Uptime + Downtime} = \frac{MTBF}{MTBF + MTTR} \quad (1)$$

Where A is the availability of physical node or VNF.

2) Availability of SFC: Considering the availability of physical node, we treat physical node and VNF running in it as a component. Thus, the availability of component is:

$$A_c = A_{n_i} A_{v_p} \quad (2)$$

Where A_c is the availability of this component, A_{n_i} is the availability of $Node_i$, and A_{v_p} is the availability of VNF_p which running in $Node_i$.

The components in SFC are organized in linear or parallel manner. We use A_{sfc} to indicate the availability of the two components. A_{c_i} and A_{c_j} indicate the availability of component i and j, respectively. The availability is as follows:

- *Linear*: For linear manner, each component has to be available at the same time. Thus, the availability of two components is:

$$A_{sfc} = A_{c_i} A_{c_j} \quad (3)$$

- *Parallel*: The parallel way can be seen as the parallel of working path and backup path. At least one of the two paths is available to guarantee the overall availability. Therefore, the availability of two components is:

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

D. Backup Model

The backup model can guarantee the availability of SFC in the event of hardware or VNF failure. However, the system is unavailable when hardware and VNF failures occur at the same time. In this section, we discuss four backup models. Firstly, we describe the VNF, Path and PV backup model. Then, we discuss our proposed JPV backup model.

We use A_{sfc} to indicate the availability of SFC. A_{n_i}, A_{n_j} and A_{n_b} indicate the availability of $Node_i$, $Node_j$ and backup $Node_b$, respectively. 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 $Node_b$, and so is θ_i and θ_j.

For simplicity, we assume that the availability of VNF_p and VNF_q is independent, and the availability of physical node and VNF running in it is also independent.

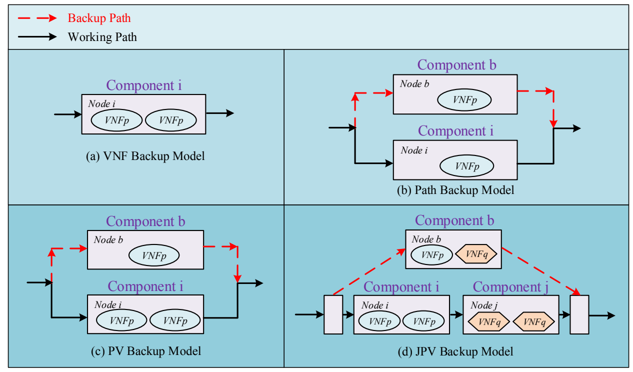

Figure 1. Backup Models: a) VNF; b) Path; c) PV; d) JPV.

1) VNF Backup Model: As Figure 1(a) shows, there are two VNF_p running in $Node_i$. The availability of SFC can be guaranteed by ensuring that the physical node is available and at least one of the VNFs running in it is available. Therefore, the availability of SFC can be shown as:

$$A_{sfc} = \prod (A_{n_i}(1 - (1 - A_{v_p})^{\theta_i})) < \prod A_{n_i} \quad (5)$$

There are only one VNF cost and no link cost in one component. And we can observe that the availability of SFC is limited by the availability of physical node.

2) Path Backup Model: As Figure 1(b) shows, one VNF_p is in the backup path while another VNF_p is in the working path. The SFC is available when at least one of the paths is available. Therefore, the availability of SFC is:

$$\begin{aligned} A_i &= A_{n_i} A_{v_p}, A_b = A_{n_b} A_{v_p} \\ A_{sfc} &= \prod (A_i + A_b - A_i A_b) < \prod A_{v_p} \end{aligned} \quad (6)$$

In this model, each component adds a backup path. The extra resources used are two links and one VNF. We can conclude that the availability of VNF is a bottleneck. In addition, we use a backup path to guarantee the availability of SFC. However, sometimes two physical nodes are not directly connected. Therefore, we need to use multiple physical links, which increases resource consumption.

3) PV Backup Model: From Eqs. 5 and 6, we can conclude that when one of Path and VNF backup achieve the highest performance, the other will become the bottleneck. Therefore, we describe PV (Path-VNF) backup model [8], which combines Path and VNF backup. One VNF_p is in the backup path and another two VNF_p are in the working path. Thus, the availability of SFC is:

$$\begin{aligned} A_i &= A_{n_i}(1 - (1 - A_{v_p})^{\theta_i}) \\ A_b &= A_{n_b}(1 - (1 - A_{v_p})^{\theta_b}) \\ A_{sfc} &= \prod (A_i + A_b - A_i A_b) \end{aligned} \quad (7)$$

Where A_b indicates both the physical node and VNF running in it are available, and so is A_i. In this model, we set θ_b to 1 by default.

For resource consumption, each component in this model adds two links and two VNFs, which is more than the other two backup models.

4) JPV Backup Model: As Figure 1(d) shows, we propose a novel JPV (Joint Path-VNF) backup model designed to optimize resource consumption while guaranteeing availability. In this model, we mainly consider three components, working component i, j and backup component b. Therefore, the availability of SFC is:

$$\begin{aligned} A_i &= A_{n_i}(1-(1-A_{v_p})^{\theta_i}) \\ A_j &= A_{n_j}(1-(1-A_{v_q})^{\theta_j}) \\ A_b &= A_{n_b}A_{v_p}A_{v_q} \\ A_{sfc} &= \prod(A_iA_j + A_b - A_iA_jA_b) \end{aligned} \tag{8}$$

Where A_b indicates $Node_b$, VNF_p and VNF_q running in the backup path are all available, and so is A_i and A_j.

In our proposed JPV backup model, one component can provide backup for two components. Each component of the JPV model adds one link and two VNFs on average, which is better than PV backup model.

Then, we compare the availability in Eq. 7 and Eq. 8. For simplicity, we assume $A_b = x$, $A_i = A_j = y$, respectively. Therefore, the result of Eq. 7 minus Eq. 8 is:

$$\begin{aligned} \Delta &= (x+y-xy) - (y^2+x-xy^2) \\ &= (1-x)(1-y)y \end{aligned} \tag{9}$$

From Eq. 9, we can see that when x and y are the minimum values, Δ is the maximum value. Therefore, we set the minimum availability of VNF and the physical node so that x and y can be the minimum. The parameter values can be found in Table I.

Table I
PARAMETER VALUES WHEN Δ IS MAXIMUM.

Parameter	Value
$A_{v_p} = A_{v_q}$	0.99
$A_{n_b} = A_{n_i} = A_{n_j}$	0.999
$x = A_b$	$0.999 * 0.99 * 0.99$
$y = A_i = A_j$	$0.999 * (1-(1-0.99)^2)$
Δ	$2.29 * 10^{-5}$

Δ is so small (at the level of 10^{-5}) 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.

IV. Problem Statement

A. Problem Description

For the resource allocation in NFV, we mainly focus on two phases: SFC composition and SFC mapping.

In composition phase, the input is SFCR and output is forwarding graph. In this paper, we name the chain of VNFs composing a network service as forwarding graph.

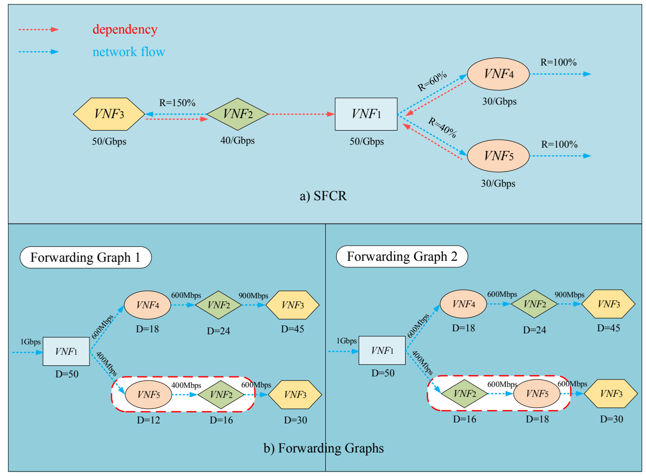

Figure 2. SFC Composition in NFV Environment.

The forwarding graph is generated based on requirements and dependencies in SFCR.

For mapping phase, the input is a forwarding graph and output is the mapping result. VNFs are mapped in the physical network based on the requirements.

1) SFC Composition: As Figure 2(a) shows, R indicates the relative flow ratio of two VNFs (e.g. R = 60% between VNF_1 and VNF_4). VNF_1 is a load balancer that divides data into two streams. 60% of incoming traffic is forwarded to VNF_4, and 40% to VNF_5. The blue dotted line indicates network flow. And the red dotted line indicates dependencies between VNFs. For example, the line from VNF_2 to VNF_1, indicating that VNF_1 must be executed before VNF_2.

As Figure 2(b) shows, there are two different forwarding graphs generated according to the requirements and dependencies in SFCR. There may be several possible VNF orders since there are no clear dependencies between some VNFs. For example, the orders of VNF_2 and VNF_5 in two forwarding graphs are different.

We assume that the initial data rate of network flow is 1 Gbps in forwarding graph 1. The data rates of two streams separated from VNF_1 are 600 Mbps and 400 Mbps, respectively. We define the relative data rate of VNF in a forwarding graph. For example, we assume that VNF_1 performs video encoding function. And VNF_1 requires the processing capabilities of a 500 MHz CPU to encode 100 MBit/s. Therefore, the relative data rate of VNF_1 is 500 * 100 = 50/Gbps and the total data rate is D = 50/Gbps * 1Gbps = 50.

By calculating the total data rate of each VNF, we can conclude that the total data rates in different forwarding graphs are different. As Figure 2(b) shows, the data rates of VNF_5 and VNF_2 are different in the red dotted box. Therefore, depending on the requirements and dependencies in SFC, different VNF orders may result in different total data rates.

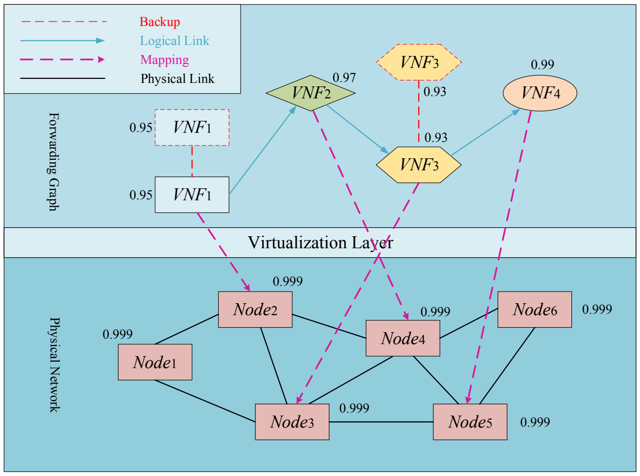

Figure 3. SFC Mapping in NFV Environment.

Considering the impacts of different forwarding graphs on network service performance, we propose a priority-based algorithm to optimize the composition and mapping of SFC. We convert the VNF dependencies to priorities, and we can get all of the forwarding graphs. Then, we can get a forwarding graph with the optimal performance by calculating the total data rate.

2) SFC Mapping: As Figure 3 shows, the SFC is mapped in NFV environment. In mapping phase, VNFs in the forwarding graph are mapped in the physical network. In this paper, we first map the working VNFs. Then, we use the backup model to provide backup for SFC and map redundant VNFs in the physical network.

As Figure 3 shows, there are four working VNFs in the forwarding graph, each with availability higher than 0.9. The availability of SFC is 0.84843 when hardware failures are ignored. However, the availability of SFC is 0.84504 when we assume the availability of physical node is 0.999. Thus, the availability of physical node should be considered.

Even if the availability of each VNF is high, the availability of SFC may become low. Therefore, it is necessary to provide backup for SFC to guarantee availability. We provide backup for VNF_1 and VNF_3, as the red dotted line indicates in Figure 3. After providing backup, the availability of SFC is 0.93131. Compared with 0.84504 before backup, it is a big improvement. It is clear that a suitable backup model can guarantee the availability of SFC efficiently.

B. Problem Formulation

In this subsection, we formulate the service chain composition and mapping problem considering availability guarantee and resource optimization. A summary of used notations is found in Table II.

Firstly, we need to meet the availability requirements of SFCR. As we mentioned in Section III, SFC is divided into several components in backup model. Therefore, the availability constraints are:

Table II
BASIC NOTATIONS USED THROUGHOUT THIS PAPER.

Symbol	Definition
Network	
N^r	the set of physical nodes. r is physical network.
C^r	the set of components in r.
n_i^r, n_j^r	two physical nodes in r.
$l_{ij}^r = (n_i^r, n_j^r)$	the physical links between n_i^r and n_j^r.
SFC	
$\mathbb{SR}$	the set of SFCRs, $s \in \mathbb{SR}$ is an SFCR.
V^s	the set of VNFs in s.
v_p^s, v_q^s	two VNFs in 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 s.
$A_{c_i}^s$	the availability of component c_i in s.
Resource	
$cpu_{v_p^s}$	the CPU consumption of v_p^s.
$mem_{v_p^s}$	the memory consumption of v_p^s.
$bw_{l_{pq}^s}$	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.
$C_{l_{ij}^r}^{bw}$	the bandwidth capacity between n_i^r and n_j^r.
Variables	
$\alpha_{v_p^s \subset n_i^r}$	whether v_p^s is mapped in n_i^r.
$\beta_{l_{pq}^s \subset l_{ij}^r}$	whether logic link l_{pq}^s is mapped in physical link l_{ij}^r.

$$\prod_{i=1}^{|C^r|} A_{c_i}^s \geq A^s \tag{10}$$

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

$$\alpha_{v_p^s \subset n_i^r} = \begin{cases} 1 & \text{if } v_p^s \text{ is mapped in } n_i^r \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

We take CPU, memory and bandwidth constraints into consideration. In detail, the CPU consumption of all VNFs placed in the same physical node cannot exceed the CPU capacity of this physical node. Therefore, we formulate the CPU constraints as:

$$\sum_{s=1}^{|\mathbb{SR}|} \sum_{p=1}^{|V^s|} cpu_{v_p^s} \cdot \alpha_{v_p^s \subset n_i^r} \leq C_{n_i^r}^{cpu} \tag{12}$$

Similarly, the memory consumption of all VNFs placed in the same physical node cannot exceed the memory capacity of this physical node. So the memory constraints are:

$$\sum_{s=1}^{|\mathbb{SR}|} \sum_{p=1}^{|V^s|} mem_{v_p^s} \cdot \alpha_{v_p^s \subset n_i^r} \leq C_{n_i^r}^{mem} \tag{13}$$

Then we use $\beta_{l^s_{pq} \subset l^r_{ij}}$ to describe whether logical link l^s_{pq} is mapped in physical link l^r_{ij}:

$$\beta_{l^s_{pq} \subset l^r_{ij}} = \begin{cases} 1 & \text{if } l^s_{pq} \text{ is mapped in } l^r_{ij} \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

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

$$\sum_{s=1}^{|\mathbb{SR}|} \sum_{p=1}^{|V^s|} bw_{l^s_{pq}} \cdot \beta_{l^s_{pq} \subset l^r_{ij}} \leq C^{bw}_{l^r_{ij}} \tag{15}$$

In this paper, our proposed solutions are designed to reduce resource consumption while guaranteeing availability. In detail, reducing the number of used physical nodes and physical links:

$$\begin{aligned} \min \sum_{s=1}^{|\mathbb{SR}|} \sum_{p=1}^{|V^s|} (\alpha_{v^s_p \subset n^r_i} + \beta_{l^s_{pq} \subset l^r_{ij}}) \\ s.t. \quad Eqs.\ 10 \quad to \quad 15 \end{aligned} \tag{16}$$

Therefore, we propose the JPV model to reduce resource consumption while guaranteeing availability and design the priority-based algorithm to reduce resource consumption.

V. Proposed Algorithm

In this section, we describe the priority-based algorithm consisting of composition phase and mapping phase. Then, we analyze the time complexity of this algorithm.

A. Composition Phase

Algorithm 1: Composition Phase

```
   Input: The SFCR s: SR^s = (V^s, L^s, A^s);
   Output: The optimal forwarding graph: FG_opt;
1  Initialize: The forwarding graph group: FG_grp = ∅;
2  foreach VNF_i in SR^s do
3  |  Convert dependency of VNF_i to priority p_i^s;
4  |  Add VNF_i with all possible orders into FG_grp;
5  end
6  foreach FG_i in FG_grp do
7  |  if Total data rate of FG_i is less than FG_opt then
8  |  |  Update FG_opt;
9  |  end
10 end
11 return FG_opt;
```

In Algorithm 1, the input is SR^s and output is the optimal forwarding graph FG_{opt}. We initialize FG_{grp} to store all of the forwarding graphs generated.

At line 3 in Algorithm 1, we convert each VNF dependency to VNF priority. For example, the VNF priority in Figure 2(a) is shown in Table III:

Table III
VNF Priority.

VNF	Priority
VNF_1	High
VNF_2	Mid
VNF_3	Low
VNF_4, VNF_5	Mid or Low

Then we add all possible orders of each VNF into FG_{grp} (line 4 in Algorithm 1). At line 6 in Algorithm 1, we traverse all of the forwarding graphs in FG_{grp} to get an optimal forwarding graph FG_{opt} as the input of mapping phase.

B. Mapping Phase

In Algorithm 2, the input is an optimal forwarding graph FG_{opt} and output is MS_{res}, which is used to store the mapping result.

Algorithm 2: Mapping Phase

```
   Input: The optimal forwarding graph: FG_opt;
   Output: The mapping result: MS_res;
1  Initialize: The initial mapping result: MS_res = ∅;
2  foreach VNF_i in FG_opt do
3  |  foreach Node_i in G_pm do
4  |  |  if Node_i can hold VNF_i then
5  |  |  |  Add mapping result into MS_res;
6  |  |  |  break;
7  |  |  end
8  |  end
9  end
10 Calculate A_sfc using Eqs. 2 and 10;
11 if A_sfc ≥ A^s then
12 |  return MS_res;
13 end
14 foreach VNF_i and VNF_j in FG_opt do
15 |  foreach Node_b in PN^r do
16 |  |  if Node_b can hold VNF_i and VNF_j then
17 |  |  |  Add mapping result into MS_res;
18 |  |  |  break;
19 |  |  end
20 |  end
21 |  Update A_sfc using Eq. 8;
22 |  if A_sfc ≥ A^s then
23 |  |  return MS_res;
24 |  end
25 end
26 if A_sfc < A^s then
27 |  Return A_sfc and ask whether A^s can be reduced;
28 end
```

In Algorithm 2, each VNF in the forwarding graph needs to be mapped in physical nodes (line 2 in Algorithm 2). If one VNF can be successfully mapped, we add the mapping result into MS_{res} (line 4-5 in Algorithm 2). After that, the traversal will be exited (line 6 in Algorithm 2) and the mapping process of the next VNF will start.

After mapping the working VNFs, we calculate the availability of SFC by using Eqs. 2 and 10 (line 10 in Algorithm 2). If the availability of SFC is not less than the availability requirements, the mapping phase ends (line 11-13 in Algorithm 2). However, the availability of SFC often does not meet the requirements. Therefore, the backup process is required (line 14-25 in Algorithm 2).

During the backup process, we use the JPV backup model. At line 14 in Algorithm 2, we provide backup for two VNFs each time. We provide backup for two VNFs with the lowest availability since it can significantly improve the availability of SFC [6]. If two VNFs can be successfully mapped, we add the mapping result into MS_{res} (line 16-17 in Algorithm 2). Then we update the availability of SFC by using Eq. 8 (line 21 in Algorithm 2). Similarly, if the availability of SFC is not less than the availability requirements, SFC meets the requirements and mapping phase ends (line 22-24 in Algorithm 2). Otherwise, we repeat the above operations until the availability of SFC meets the requirements.

However, the availability of SFC may still not meet availability requirements even though all backup are provided (line 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.

C. Complex Analysis

As we mentioned in Section III, δ, ζ and η indicate the number of physical nodes, VNFs and logical links, respectively.

For composition phase, *foreach* (line 2 in Algorithm 1) runs ζ times. At line 6 in Algorithm 1, the number of FG_{grp} is too small to be ignored. However, the calculation of total data rate runs ζ times (line 7 in Algorithm 1). Thus, the time complexity of composition phase is:

$$\zeta + \zeta = 2\zeta \tag{17}$$

In mapping phase, we use BFS (Breadth First Search) to traverse FG_{opt}. Therefore, *foreach* (line 2 in Algorithm 2) runs (ζ + η) times. At line 3 in Algorithm 2, *foreach* runs δ times. At line 10 in Algorithm 2, the Eqs. 2 and 10 run ζ times. In backup process, *foreach* (line 14 in Algorithm 2) runs $\zeta/2$ times. At line 15 in Algorithm 2, *foreach* runs δ times. And the Eq. 8 runs $\zeta/2$ times (line 21 in Algorithm 2). Thus, the time complexity of mapping phase is:

$$(\zeta + \eta) \cdot \delta + \zeta + \frac{\zeta}{2} \cdot (\delta + \frac{\zeta}{2}) = \eta\delta + \zeta + \frac{3}{2}\zeta\delta + \frac{\zeta^2}{4} \tag{18}$$

From Eqs. 17 and 18, the time complexity of composition and mapping is at the level of O(ζ) and O($\eta\delta + \zeta\delta + \zeta^2$), respectively. Therefore, we can conclude that the time complexity of priority-based algorithm is at the level of O($\eta\delta + \zeta\delta + \zeta^2$).

VI. Performance Evaluation

A. Simulation Setup

In this simulation, we implement our proposed JPV backup model and priority-based algorithm. We mainly focus on the following metrics: total data rate, acceptance ratio, max availability and resource consumption. In order to make it easier to observe the performance of our proposed JPV backup model, we compare it with VNF, Path and PV [8] backup model, as described in Section III.

We evaluate the performance using a laptop of windows 10 with 2.2 GHz Intel Core i5 processor and 8 GB memory. We implement the backup models and mapping algorithms in Java based on Alevin [19], a wider simulation environment for VNF resource allocation. All of the statistics in this simulation are average results and almost unaffected by the accidental events. Here are the parameters:

1) SFC Request: In the simulation, each SFCR consists of two to six VNFs. Each VNF can provide one network function, and it requires three types of physical resources. There are eight types of VNFs with availability between [0.99, 0.9999]. The logic link has a bandwidth demand between [10, 30]. And the computing and memory requirements of VNF are between [10, 20] and [5, 10], respectively. By referring to Google Apps [23] and other literatures [8], [14], we divide the availability requirements of SFC into four levels among {0.99, 0.999, 0.9999, 0.99999}.

2) Physical Network: In this paper, we use Barabasi-Albert generator [10] to generate random physical network topologies [24]. The physical network contains three types of resources including bandwidth, computing and memory, with the capacity between [50, 100]. There are 200 physical nodes in the default network topology. The availability of physical node is distributed within [0.999, 0.99999].

B. Total Data Rate

In this subsection, we evaluate the total data rate of the priority-based algorithm (PBA) and random-based algorithm (RBA). The difference between PBA and RBA is whether the composition phase is used. As Figure 4(a) shows, the more VNFs in SFCR, higher total data rate. The total data rate of PBA is smaller than RBA under the same availability level. Therefore, we conclude that the composition phase in PBA can reduce the total data rate effectively.

C. Acceptance Ratio

We evaluate the acceptance ratio of four backup models, as shown in Figure 4(b). When the availability level is two 9s, all of the backup models are close to 1. When three 9s, the three 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 five 9s, the acceptance ratios of JPV are 0.03, while PV is 0.04. In this case, we can think that PV and JPV

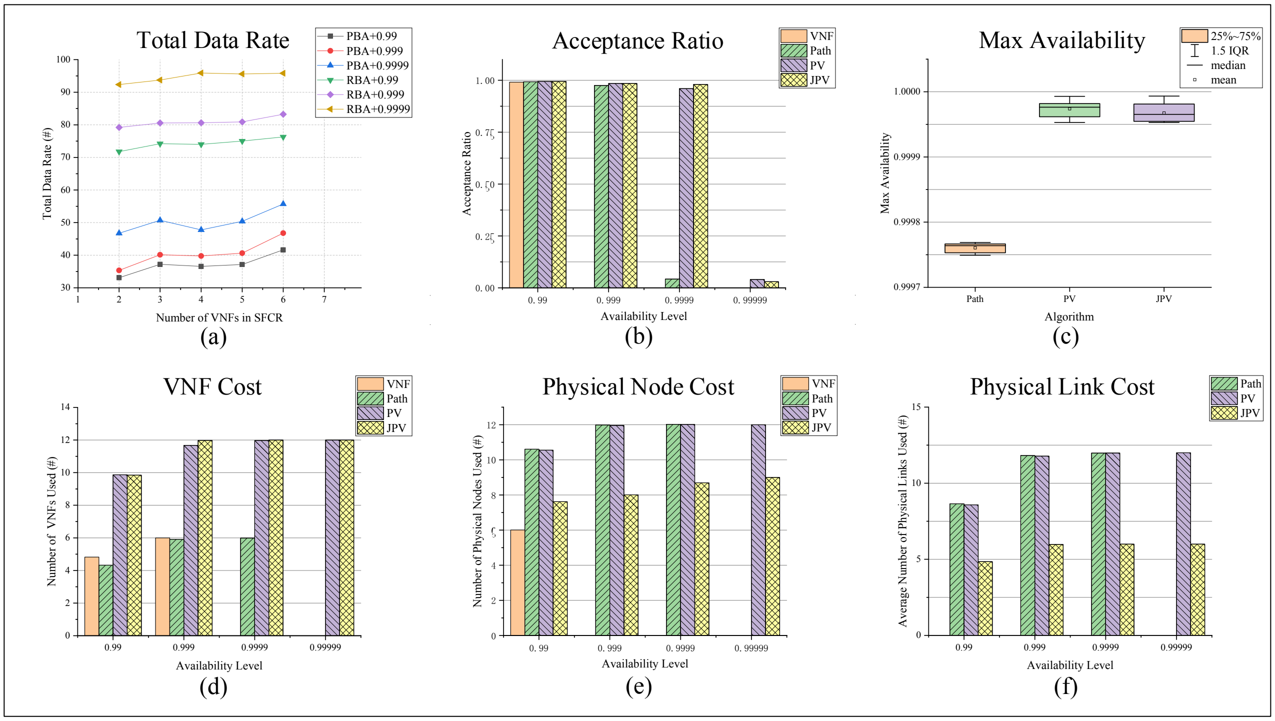

Figure 4. Evaluation: a) Total Data Rate; b)Acceptance Ratio; c) Max Availability; d) VNF Cost e) Physical Node Cost; f) Physical Link Cost.

cannot meet five 9s. In summary, PV and JPV perform the same in terms of acceptance ratio.

D. Max Availability

As Figure 4(c) shows, we are mainly 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 Figure 4(c). The Path backup model is 0.99976, which satisfies three 9s. The availability of PV is 0.999973 while JPV is 0.999967. We conclude that PV and JPV backup models can meet four 9s. It shows that the impact of Δ (mentioned in Section III) between PV and JPV on availability improvement can be ignored.

E. VNF Cost

As Figure 4(d) shows, partial data of VNF and Path backup model are 0. It means that the two backup models cannot meet the availability requirements in these cases. Therefore, we mainly discuss the VNF costs of PV and JPV backup model. We can observe that as the availability level increases, the VNF costs of two backup models also increase. It means that the backup model needs more redundant VNFs to achieve higher availability. As we discussed in Section III, the average VNF costs of PV and JPV model are the same. Similarly, the VNF costs of JPV model under different availability levels are basically the same as PV backup model, as shown in Figure 4(d).

F. Physical Node Cost

As Figure 4(e) shows, the physical node costs of backup models increase as the availability level increases, although it is not obvious. It shows that all backup models require more physical nodes to achieve higher availability. We can also observe that the physical node costs of Path and PV are close, and they are obviously higher than JPV under the same availability level. This means that JPV backup model can achieve the same availability as the other backup models with less physical nodes.

G. Physical Link Cost

In this subsection, we compare the link costs of other three backup models since VNF backup model has no link cost, as shown in Figure 4(f). We can observe that the link costs of backup models increase as the availability level increases. It means that the backup model needs more links to improve availability. As Figure 4(f) shows, we can conclude that the link costs of Path and PV backup model are basically the same. Compared with Path and PV backup model, the link cost of JPV backup model is reduced by nearly 50%.

VII. Conclusion

In this paper, we focus on the composition and mapping of SFC considering availability guarantee and resource optimization. We define an availability model that takes both physical node and VNF failures into consideration. And we propose a JPV backup model and priority-based algorithm to guarantee availability and optimize resource consumption. The simulation results show that JPV backup model and priority-based algorithm can reduce resource consumption while guaranteeing availability.

We also recognize the limitations of our research. Large-scale service chain composition problems often use heuristic algorithms but increase the time to get a sub-optimal solution. Therefore, the service chain composition problem in NFV-enabled networks requires an efficient solution. In future work, we mainly focus on the optimization of service chain composition.

ACKNOWLEDGMENT

This work was supported in part by the National Key Research and Development Program of China No.2018YFB1003804, in part by the National Natural Science Foundation of China No.61772479, and in part by the BUPT Excellent Ph.D. Students Foundation under Grant CX2019214.

REFERENCES

[1] Network functions virtualisation (nfv). [Online]. Available: https://portal.etsi.org/Portals/0/TBpages/NFV/Docs/NFV_White_Paper3.pdf

[2] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. D. Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Communications Surveys and Tutorials*, vol. 18, pp. 236–262, 2016.

[3] 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 *SIGCOMM*, 2018.

[4] P. Gill, N. Jain, and N. Nagappan, "Understanding network failures in data centers: measurement, analysis, and implications," in *SIGCOMM*, 2011.

[5] Opnfv. [Online]. Available: https://wiki.opnfv.org/display/availability

[6] J. Fan, Z. Ye, C. Guan, X. Gao, K. Ren, and C. Qiao, "Grep: Guaranteeing reliability with enhanced protection in nfv," in *HotMiddlebox@SIGCOMM*, 2015.

[7] J. Fan, M. Jiang, and C. Qiao, "Carrier-grade availability-aware mapping of service function chains with on-site backups," *2017 IEEE/ACM 25th International Symposium on Quality of Service (IWQoS)*, pp. 1–10, 2017.

[8] J. Kong, I. Kim, X. Wang, Q. Zhang, H. C. Cankaya, W. Xie, T. Ikeuchi, and J. P. Jue, "Guaranteed-availability network function virtualization with network protection and vnf replication," *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, pp. 1–6, 2017.

[9] M. Mechtri, C. Ghribi, and D. Zeghlache, "A scalable algorithm for the placement of service function chains," *IEEE Transactions on Network and Service Management*, vol. 13, pp. 533–546, 2016.

[10] M. T. Beck and J. F. Botero, "Scalable and coordinated allocation of service function chains," *Computer Communications*, vol. 102, pp. 78–88, 2017.

[11] L. Wang, Z. Lu, X. Wen, R. Knopp, and R. Gupta, "Joint optimization of service function chaining and resource allocation in network function virtualization," *IEEE Access*, vol. 4, pp. 8084–8094, 2016.

[12] D. Li, P. Hong, K. Xue, and J. Pei, "Virtual network function placement considering resource optimization and sfc requests in cloud datacenter," *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, pp. 1664–1677, 2018.

[13] Etsi gs nfv-rel 001 v1.1.1. [Online]. Available: https://portal.etsi.org/Portals/0/TBpages/NFV/Docs/NFV_White_Paper3.pdf

[14] J. Fan, C. Guan, Y. Zhao, and C. Qiao, "Availability-aware mapping of service function chains," *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, pp. 1–9, 2017.

[15] 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," *2015 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–7, 2015.

[16] J. Sun, G. Zhu, G. Sun, D. Liao, Y. Li, A. K. Sangaiah, M. Ramachandran, and V. I. Chang, "A reliability-aware approach for resource efficient virtual network function deployment," *IEEE Access*, vol. 6, pp. 18 238–18 250, 2018.

[17] J. Kang, O. Simeone, and J. Kang, "On the trade-off between computational load and reliability for network function virtualization," *IEEE Communications Letters*, vol. 21, pp. 1767–1770, 2017.

[18] 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 Transactions on Network and Service Management*, vol. 14, pp. 554–568, 2017.

[19] J. Gil-Herrera and J. F. Botero, "Resource allocation in nfv: A comprehensive survey," *IEEE Transactions on Network and Service Management*, vol. 13, pp. 518–532, 2016.

[20] A. Fischer, J. F. Botero, M. T. Beck, H. de Meer, and X. Hesselbach, "Virtual network embedding: A survey," *IEEE Communications Surveys and Tutorials*, vol. 15, pp. 1888–1906, 2013.

[21] P. A. Kullstam, "Availability, mtbf and mttr for repairable m out of n system," *IEEE Transactions on Reliability*, vol. R-30, pp. 393–394, 1981.

[22] B. Liu, L. Cui, S. Si, and Y. Wen, "Performance measures for systems under multiple environments," *IEEE/CAA Journal of Automatica Sinica*, vol. 3, pp. 90–95, 2016.

[23] Google apps service level agreement. [Online]. Available: https://gsuite.google.com/intl/en/terms/sla.html

[24] B. I. A. McInnes, J. S. McBride, N. J. Evans, D. D. Lambert, and A. S. Andrew, "Emergence of scaling in random networks," 1999.