A Service Function Chain Deployment Scheme Based on Heterogeneous Backup

Jichao Xie
National Digital Switching System Engineering & Technological Research Center
Zhengzhou, China
e-mail: 912104210329@njust.edu.cn

Peng Yi
National Digital Switching System Engineering & Technological Research Center
Zhengzhou, China
e-mail: 15238363582@139.com

Zhen Zhang
National Digital Switching System Engineering & Technological Research Center
Zhengzhou, China
e-mail: Zhanzhen2096@163.com

Chuanhao Zhang
Public Security Technology Department
Rail Police College
Zhengzhou, China
e-mail: zhangchuanhao@rpc.edu.cn

Yunjie Gu
National Digital Switching System Engineering & Technological Research Center
Zhengzhou, China
e-mail: lizardwhite@163.com

***Abstract*—Building a service function chain (SFC) based on Virtual Network Functions (VNFs) greatly improves the flexibility of network service deployment. However, compared to highly reliable carrier-grade proprietary hardware, VNF is facing a series of failure risks. Although the existing backup methods can solve the problem of VNF failure to some extent, malicious attacks can't be stopped totally because there is no consideration of the homogeneity of the original node and the backup nod. In this paper, we propose a heterogeneous backup deployment scheme. It ensures the heterogeneity of backup server nodes and VNF nodes with the original one during backup. We designed and implemented the corresponding deployment algorithm. Simulation results show that compared with the backup method without considering heterogeneity, our method increases attacker's attack time cost by 13.2% under the cost that the request acceptance rate decreases by 4.81%.**

Keywords-NFV; VNF; SFC; reliability; backup; homogeneity; heterogeneity

I. INTRODUCTION

General network services require that traffics be processed by a series of network functions (NFs) in a specific order. These specific ordered network function sequences are called service function chains (SFCs) [1]. Traditional network services provision method requires deploying a large number of proprietary hardware in the network, such as firewalls, intrusion detection systems (IDS) and proxy servers. These proprietary hardwares play an indispensable role in improving network performance and security, but it also bring the network with poor scalability, difficulty in flexible deployment, and high maintenance and operation costs [2].

Network function virtualization (NFV) technology uses virtualization technology and cloud computing technology to realize relevant NFs in the form of software. Virtual network functions (VNFs) run in the form of virtual machines on commodity servers, switches and storages [3]. Building SFCs based on VNFs greatly improves the efficiency and flexibility of network service deployment. At present, major telecom operators at home and abroad are actively promoting the deployment of NFV technology. AT&T plans to virtualize 75% of its large networks by the end of 2020 [4]. China Mobile, China Telecom and China Unicom are also actively promoting the application of NFV technology.

However, compared to highly reliable carrier-grade network equipment, the vulnerability of software VNFs brings certain reliability risks to the network [5]. Factors that cause VNFs to fail are complex and diverse, such as hardware failures and software failures. In addition, NFV's open market environment, third-party vendors' hardware and software solutions, make it more difficult for hardware and software to completely control and eliminate backdoors, traps and loopholes. These potential unknown risks make VNF face a series of uncertain threats. The failure of any NF node will cause the failure of the related SFC, resulting in service interruption, data loss and waste of resources.

In order to deal with the failure of network function nodes, the existing research mainly uses redundant backup methods to enhance the reliability of SFCs. In engineering practice, authors in [6] implemented a VNF deployment system with automatic recovery capabilities in a real engineering experiment environment. When a single VNF

978-1-5386-7633-2/18/$31.00 ©2018 IEEE

node failure occurs, failure recovery is achieved by rerouting the failed node's traffic to the backup node.

In theoretical research, [7] [8] adopted the method of backing up the entire SFC to improve service reliability. In [8], an iterative deployment algorithm was designed to iteratively add backup SFC link to SFC primary link until the required reliability requirements or the maximum number of backup SFC is reached. However, backup of the entire SFC has excessive resource overhead. In [9], [10], [11], [12], targeted backup strategies were adopted. In [9], authors improved reliability by providing redundancy for VNFs in different server clusters and designed a variable domain search deployment algorithm. However, the chaining of VNFs was not considered and only the reliability deployment of a single virtual network function was considered. In [10], a backtracking deployment algorithm based on breadth-first search was designed to solve the problem of node and link failures in the NFVI infrastructure. But this method selects backup nodes and links randomly, which has certain defects. Authors in [11] considered that backup VNFs can be shared by different VNFs. And designed a screening model that iteratively selects the most cost-efficient node for backup to improve overall SFC reliability. Avoid the excessive backup and resource utilization issues. In [12], the authors considered that the reliability of the underlying nodes are different. They improved the reliability of SFCs by migrating VNFs from low-reliability servers to highly reliable servers and providing redundant backups for some VNFs. However, this method will cause a certain load imbalance problem. The business of high reliability nodes' is too heavy, Once one of them fails, it will have a wider impact.

For the failure of VNFs problem, the current SFC reliability deployment schemes have a good effect in dealing with random failure of VNFs. However, due to the lack of consideration of homogeneity among nodes, there is a huge risk in the face of malicious attacks. The software of the backup VNFs are the same as the original VNFs. The backup physical servers and the original physical servers are also identical (Have the same kind of CPU or operating system), which makes the backup nodes and the original nodes have common defects. Malicious attackers do not need to spend a lot of time retrieving and detecting system defects after they have successfully attacked related nodes and then can perform a new attack on the backup nodes that cause the VNFs to fail again. That will cause long-term failure of related SFCs and disruption of related services.

Considering the reliability problems faced by SFC, we should take the heterogeneity of nodes into account, because heterogeneous redundant backups have greater gains in improving SFC reliability. The heterogeneity of the backup nodes can be considered from three aspects: hardware, operating system and application software.

Hardware heterogeneity: Such as Intel CPU and AMD CPU. Some Intel CPUs have Meltdown and Spectre vulnerabilities due to the design defects which cause a great threat to user privacy and device security. But AMD CPUs have different processor architecture, only the Spectre has a certain impact on its products.

Operating system heterogeneity: Such as Microsoft's Windows Server operating systems and some operating systems based on the Linux kernel (Ubuntu, Red Hat, Fedora, CentOS). There are differences in the vulnerabilities.

Application software heterogeneity: There are also different vulnerabilities between the three Web servers of Apache, Nginx, and Internet Information Server (IIS)

This paper introduces the idea of heterogeneity into the reliable deployment of SFC. A SFC deployment algorithm based on heterogeneous backup was designed to address the uncertain failures and uncertain threats faced by SFC. The rest of the paper is organized as follows: Section II describes the network model and heterogeneous backup methods. Section III presents the heterogeneous backup deployment model and algorithm implementation; Section IV introduces the results of relevant simulation experiments. Finally, Section V presents the conclusion and future work.

II. NETWORK MODEL AND HETEROGENEOUS BACKUP SCHEME

A. *Network Function Virtualization Model*

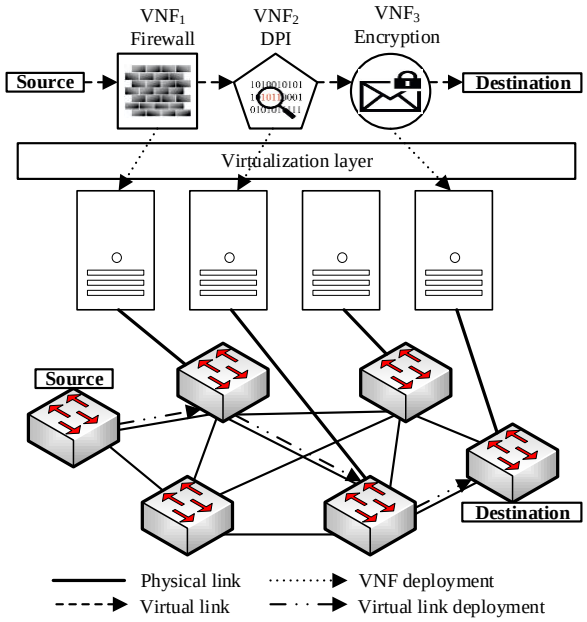

Figure 1. Service Function Chain Deployment Model

The SFC deployment model based on NFV is shown in Figure 1. A network service traffic needs to be processed by firewalls, deep packet inspection (DPI) functions, and encryption service functions in turn. The essence of SFC deployment are the deployment of network functions and the guidance of traffic. According to the request information of the network service and the status information of the underlying network resources, the deployment of VNFs and virtual links have been accomplished by the management and scheduling module. The VNFs form the corresponding SFCs to provide services for users according to the order of execution specified in the service flow.

B. Substrate Network Model

1) Physical network

We use an undirected graph $\bar{G}=(\bar{S},\bar{L})$ to represent the substrate network, where $\bar{S}$ and $\bar{L}$ represents the switch collection and the physical link collection, respectively. The set $\bar{N}$ represents these servers that can deploy VNFs. Let, s , l and n denotes the total number of switches, physical links, and servers in the underlying network, respectively.

We define the matrix $\boldsymbol{B}_{s\times s}$ as the switch adjacency matrix which indicates the adjacency relationship between all switches. And $B_{i,j}\in R^{+}$ represents the bandwidth of the communication link from the switch node $i\in\bar{S}$ to the switch node $j\in\bar{S}$.

Let, binary matrix $\boldsymbol{H}_{n\times s}$ denotes the connection matrix between servers and switches, where indicates that whether the server node $i\in\bar{N}$ is connected to the switch node $j\in\bar{S}$. And each server can only connect to one switch node. There is $\sum_{j\in\bar{S}}H_{i,j}=1 \quad \forall i\in\bar{N}$.

The set K represents these resources that a server node can provide (e.g., CPU, memory, storage, hardware acceleration resources, etc.). Let k indicates the total number of available resource types.

Define matrix $\boldsymbol{C}_{n\times k}$ as the server resource capacity matrix which represents the capacity of various types of resources. The element $C_{i,j}\in R^{+}$ represents the available resource capacity of $j\in K$ on the server node i .

Let T represents the set of server node types (Have different CPUs or different operating systems are different types of heterogeneous servers.). The total number of server types in the network is represented by t . The heterogeneous server types are denoted by $1,2,\cdots t$.

Define the row matrix $\boldsymbol{N}_{1\times n}$ as the server type vector, which indicates the composition of server types in the network, where element $N_{1,i}\in T=\{1,2,\cdots t\}$ represents the type of the server.

2) Virtual Network Function

Let, z to represent the total number of different VNFs (e.g., Firewall, IDS, Encryption, Proxy, etc.) that can be provided in the entire network. And the total number of VNFs types are denoted by P .

$1,2,3,\cdots 1+z,2+z,3+z,\cdots 1+2z,2+2z,3+3z\cdots p$ represent Firewall_1, IDS_1, Encryption_1... heterogeneous Firewall_2, heterogeneous IDS_2, heterogeneous Encryption_2... heterogeneous Firewall_3, heterogeneous IDS_3, heterogeneous Encryption_3, respectively. Each VNF has a deployment resource demand that is related to the size of the processing traffic. Define matrix $\boldsymbol{Q}_{p\times k}$ as the VNF resource demand coefficient matrix. Element $Q_{i,j}$ indicates the quantity of j type resources required by i type VNF to process 1 unit bandwidth traffic.

Define the binary matrix $\boldsymbol{S}_{n\times p}$ represent the VNF type set that a sever nodes can support. If $S_{i,j}=1$, indicates that the server node i can support the deployment of j type VNF.

3) SFC request

We assume that the network operator continues to receive SFC requests. Let, R represents the SFC requests. A SFC request is represented by a 5-tuple $r=<\bar{u}^r,\bar{v}^r,\beta^r,\tau^r,\psi^r>$,where $\bar{u}^r,\bar{v}^r$ denote the ingress and egress switches, respectively. β^r indicates the amount of traffic that this request needs to process. τ^r indicates the duration of the request. ψ^r indicates the sequence of VNFs which the traffic must pass (e.g. Firewall→IDS→Encryption). And let m indicates the total number of VNFs in this SFC request.

Define the row vector $\boldsymbol{T}_{1\times m}$ to represent the VNF type vector of a SFC request. The element $T_{1,i}\in\{1,2,\cdots p\}$, $i\in\{1,2,\cdots m\}$ indicates the type of the i th VNF in the request.

We represent the VNF sequence as a directed graph $G^r=(N^r,L^r)$, where N^r represents the desired set of virtual nodes (ingress switches, VNFs, egress switches), and L^r indicates the virtual link connecting these virtual nodes.

C. Heterogeneous Backup Deployment Scheme

The existing redundancy backup methods have not considered the homogeneity of nodes. This paper introduces heterogeneous idea into the deployment of SFC. There are some problems such as excessive resource overhead, low resource utilization, and serious drop in mapping success rate in the backup of all VNFs of SFC. We only perform redundant backups of critical, vulnerable, and less reliable nodes. And consider the heterogeneity of nodes (heterogeneity of underlying server hardware, operating systems, and VNFs) when performing redundant backups to avoid common defects between the original node and the newly deployed node, which makes the attacker's information available to the SFC unsustainable. Even if the current attack is successful, it is still necessary to conduct system vulnerability mining again when the attack is performed again, which significantly increases the cost of the attack. The method is specifically described as follows.

Define the row vector $\boldsymbol{BK}^r_{1\times m}$ to denote the backup vector of VNF. If $BK^r_{1,i}=1$, we need to provide heterogeneous backup $VNF_i^{r,backup}$ for VNF_i^r . The backup requirements variables $BK^r_{1,i}$ can be assigned based on reliability statistics and backup strategies of the relevant VNFs.

When deploy an SFC, we perform SFC primary link mapping in the first, and then perform heterogeneous backup for VNF_i^r which has backup requirement. As shown in Figure 2, the SFC request contains three VNFs, of which VNF_2 need to be backed up. Firstly, we complete the deployment of SFC primary links on servers N4, N1, and N5.

And then complete the deployment of backup node VNF_2^{backup} on the heterogeneous server N2.

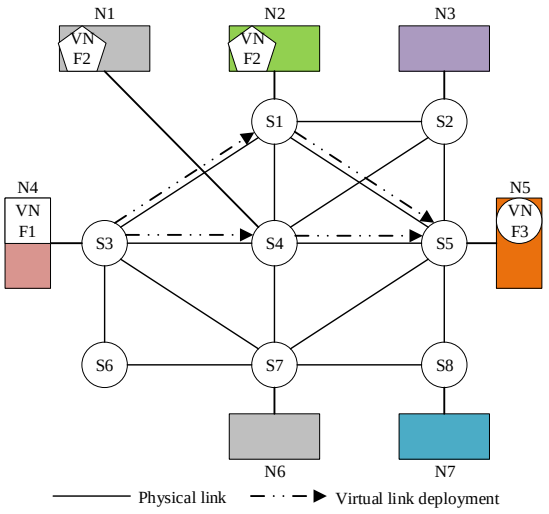

Figure 2. Heterogeneous backup

III. HETEROGENEOUS BACKUP FORMULATION AND ALGORITHM

A. *Heterogeneous Backup Deployment Model*

Define a binary matrix $\boldsymbol{F}_{m\times n}^{r}$ as the deployment relationship between VNF and server node in the rth SFC request, where $F_{i,j}^{r}\in\{0,1\}$ indicates whether the ith VNF in the rth SFC request is deployed on server node j. We allow different VNFs in the same SFC request to be mapped on the same server node.

Let, $n_i^r\in\bar{N}$ denotes the server node that the VNF_i^r deploy on, and $s_i^r\in\bar{S}$ represents the switch node which the server node $n_i^r\in\bar{N}$ directly connected to. Then, $F_{i,n_i^r}^{r}=1$ and $H_{n_i^r,s_i^r}=1$. In addition, when $i=0$, let $s_0^r=\bar{u}$, indicating the ingress switch. And when $i=m+1$, let $s_{m+1}^r=\bar{v}$, representing the egress switch.

We also define $V=\eta(u)=\{v\mid B_{u,v}>0\}\subseteq\bar{S},\ u\in\bar{S}$ as the set of switches that the switch u directly connected to.

Define a binary matrix $\boldsymbol{E}_{s\times s}^{s_i^r,s_{i+1}^r}$ as the deployment relationship between virtual link $L_{i,i+1}^{r}$ and physical link $\bar{L}_{u,v}$ from VNF_i^r to $VNF_{i+1}{}^r$, where $E_{u,v}^{s_i^r,s_{i+1}^r}\in\{0,1\}$ indicates whether the virtual link $L_{i,i+1}^{r}$ is deployed on the physical link $\bar{L}_{u,v}$.

We also define matrices $\boldsymbol{C}_{n\times k}^{rem}$ and $\boldsymbol{B}_{s\times s}^{rem}$ to the remaining resources of server nodes and the remaining bandwidth of link, respectively.

We first perform primary link deployment and then deploy the $VNF_i^{r,backup}$.

1) Primary link deployment constraints:

a) VNF deployment constraints:

$$S_{n_i^r,VNF_i^r}=1\quad\forall i\in\{1,2,\cdots,m\},VNF_i^r\in\psi^r,n_i^r\in\bar{N}\tag{1}$$

$$\sum_{i=1}^{m}Q_{VNF_i^r,k}\times\beta^r\times F_{i,n}^{r}\le C_{n,k}^{rem}\quad\forall k\in K,\forall n\in\bar{N}\tag{2}$$

$$\sum_{j\in\bar{N}}F_{i,j}^{r}=1\quad\forall i\in\{1,2,\cdots,m\}\tag{3}$$

Equation (1) ensure that the server node n_i^r must support the deployment of VNF_i^r. Equation (2) make sure that the total amount of resources used by all VNFs in the SFC request cannot exceed server nodes' available resources. Equation (3) ensure that each VNF_i^r can only be deployed on one server node.

b) Link deployment constraints:

The bandwidth resources occupied by all the virtual links in the request cannot exceed the remaining bandwidth resources of each physical link. We express this constraint as equation (4).

$$\sum_{i=0}^{m}E_{u,v}^{s_i^r,s_{i+1}^r}\times\beta^r\le B_{u,v}^{rem}\quad\forall u\in\bar{S},v\in V=\eta(u)\tag{4}$$

c) Objective function:

$$Min\left[\sum_{i\in\{0,1,\cdots,m\}}\sum_{u\in\bar{S}}\sum_{v\in V=\eta(u)}E_{u,v}^{s_i^r,s_{i+1}^r}\times\beta^r\right]\tag{5}$$

For the deployment of the same SFC, the server resources used by each VNF to deploy at any service node are the same. But the bandwidth resource consumption of the virtual link may vary depending on the VNFs' deployment server node. In order to reduce resource consumption, we take the minimum bandwidth resource consumption as our objective function, as expressed in equation (5).

2) Heterogeneous backup constraints:

Let, $n_i^{rb}\in\bar{N}$ denotes the server node that $VNF_i^{r,backup}$ deploy on, and $s_i^{rb}\in\bar{S}$ represents the switch node which the server node $n_i^{rb}\in\bar{N}$ directly connected to. Then, there are $F_{i,n_i^{rb}}^{rb}=1$ and $H_{n_i^{rb},s_i^{rb}}=1$.

When $BK_{1,i}^{r}=1,i\in\{1,\cdots,m\}$, We need to provide heterogeneous redundancy for the corresponding VNF_i^r during the deployment phase. First, we need to complete the deployment of $VNF_i^{r,backup}$ on server node $n_i^{rb}\in\bar{N}$. Secondly, complete the deployment of three (or two) virtual links between $VNF_i^{r,backup}$ and VNF_{i-1}^{r}, $VNF_i^{r,backup}$ and VNF_{i+1}^{r}, $VNF_i^{r,backup}$ and $VNF_{i-1}^{r,backup}$ (may not exist). The objective function is also to minimize link bandwidth consumption.

a) VNF deployment constraints:

$$BK_{1,i}^{r}=1\quad i\in\{1,\cdots,m\}\tag{6}$$

$$VNF_i^{r,backup}=VNF_i^r+z\times j\quad i\in\{1,\cdots,m\},j\in\{1,2\}\tag{7}$$

$$S_{n_i^{rb},VNF_i^{r,backup}}=1\quad n_i^{rb}\in\bar{N},VNF_i^{r,backup}\in P\tag{8}$$

$$n_i^{rb}\ne n_i^r,N_{1,n_i^{rb}}\ne N_{1,n_i^r}\quad n_i^{rb},n_i^r\in\bar{N}\tag{9}$$

$$Q_{VNF_i^{r,backup},k}\times\beta^r\times F_{i,n_i^{rb}}^{r}\le C_{n_i^{rb},k}^{rem}\quad\forall k\in K\tag{10}$$

$$\sum_{j\in\bar{N}}F_{i,j}^{r,b}=1\tag{11}$$

Equation (6) represents the precondition for providing a redundant backup. Equation (7) ensure that the provided redundant backup VNF is heterogeneous. Equation (8) ensure that the server node n_i^{rb} must support the deployment of $VNF_i^{r,backup}$. Equation (9) makes sure that the server node providing the backup is heterogeneous with the original node. Equation (10) ensure that the various resource capacities of server node n_i^{rb} meet the deployment requirements. Equation (11) ensure that each $VNF_i^{r,backup}$ can only be deployed on one server node.

b) Link deployment constraints:

Three (or Two) backup virtual links deployment constraint can be expressed as follow:

$$\left(E_{u,v}^{s_{i-1}^r,s_i^{rb}}+E_{u,v}^{s_i^{rb},s_{i+1}^r}+E_{u,v}^{s_{i-1}^{rb},s_i^{rb}}\right)\times\beta^r\le B_{u,v}^{rem}$$
$$\forall u\in\bar{S},v\in V-\eta(\mathrm{u}) \tag{12}$$

c) Objective function:

The objective function is also to minimize link bandwidth consumption.

$$Min\left[\sum_{u\in\bar{S}}\sum_{v\in V-\eta(\mathrm{u})}\left(E_{u,v}^{s_{i-1}^r,s_i^{rb}}+E_{u,v}^{s_i^{rb},s_{i+1}^r}+E_{u,v}^{s_{i-1}^{rb},s_i^{rb}}\right)\times\beta^r\right] \tag{13}$$

B. Algorithm Design

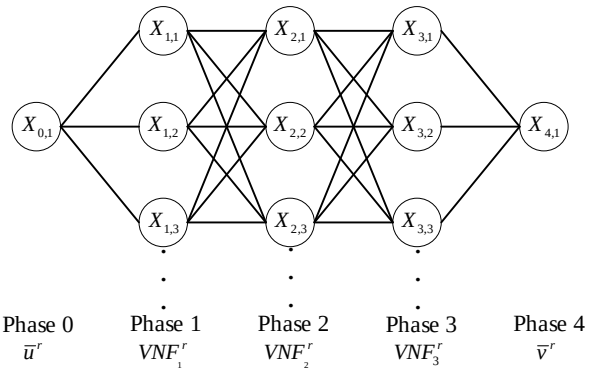

Figure 3. Multi-phase diagram of primary link deployment

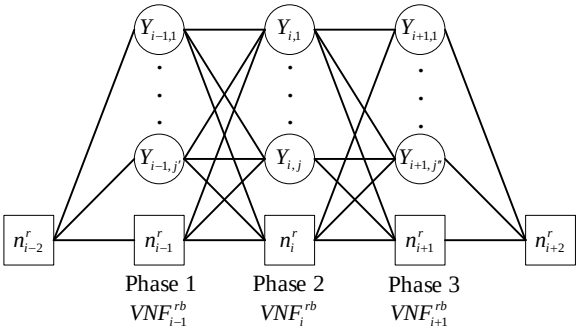

Figure 4. Multi-phase diagram of heterogeneous backup deployment

Taking the constraints of Section III A as constraints, we further designed the basic deployment algorithm based on Viterbi proposed in [13], and presented a heterogeneous backup deployment algorithm. The pseudo code of the algorithm is shown in Table I. Due to space limitations, the details of the implementation process will be described using Figure 3 and Figure 4 briefly.

TABLE I. HETEROGENEOUS BACKUP DEPLOYMENT ALGORITHM

Input: SFC Request information r, Backup requirements $BK_{1\times m}^r$, Current network resource status $C_{n\times k}^{rem}$, $B_{s\times s}^{rem}$

Output: Heterogeneous backup deployment results with minimal bandwidth consumption

1.#Primary link deployment
2.Determine the set of servers $\bar{N}_{VNF_i^r}$ that meet the deployment requirements for each VNF_i^r.
3.**For each** $i\in\{0,1,\cdots,m,m+1\}$ # Primary link deployment has m + 2 phases
4. **For each** $X_{i,j}\in\bar{N}_{VNF_i^r}$ # Traversing all the server nodes that can deploy VNF_i^r.
5. **For each** $X_{i-1,j'}\in\bar{N}_{VNF_{i-1}^r}$ # Traversing all the server nodes that can deploy VNF_{i-1}^r.
6. $\cos t^{current}=\cos t(X_{i-1,j'},X_{i,j})$ #Calculate the deployment cost between server nodes $X_{i-1,j'}$ and $X_{i,j}$
7. $\pi_{i,j,j'}=\cos t^{current}+\cos t(X_{i-1,j'})$ # Calculate the deployment cost from $X_{i-1,j'}$ to $X_{i,j}$
8. **End for**
9. Let $\cos t(X_{i,j})$ equals to the minimum value in $\pi_{i,j,j'}$ and records the relevant deployment node and link # Calculate the minimum deployment cost to reach $X_{i,j}$
10. **End for**
11.**End for**
12.The minimum value recorded in $\cos t(X_{m+1,1})$ and the corresponding nodes and links are the required minimum bandwidth occupation deployment results
13. Update the network resource status

14.#Heterogeneous backups deployment
15. $\bar{N}_{VNF_i^{r,backup}}$ Determine the set of servers $\bar{N}_{VNF_i^{r,backup}}$ that meet the deployment requirements for each VNF_i^r with backup requirements.
16.Analyze the elements in $BK_{1\times m}^r$ to determine the backup nodes, Let, y indicates the number of consecutive backup nodes. The deployment of a single backup node ($y=1$) is simple. But for continuous backup nodes ($y\ge 2$), the Multi-phase diagram are required to determine the optimal deployment results。
17.**If** $y=1$
18. **For** $Y_{i,j}\in\bar{N}_{VNF_i^{r,backup}}$
19. $\pi_{i,j}=\cos t(n_{i-1}^r,Y_{i,j})+\cos t(n_{i+1}^r,Y_{i,j})$

20. **End for**
21. Let $cost(Y_i)$ equals to the minimum value in $\pi_{i,j}$ and records the relevant deployment node and link.
22. Update the network resource status
23. **End if**
24. **If**
25. **For each** $i \in$ {y nodes that need to be backed up}
26. **For each** $Y_{i,j} \in \bar{N}_{VNF_i^{r,backup}}$
27. **For each** $Y_{i-1,j'} \in \bar{N}_{VNF_{i-1}^{r,backup}}$
28. $cost^{current} = cost(n_{i-1}^r, Y_{i,j}) + cost(n_{i+1}^r, Y_{i,j}) + cost(Y_{i-1,j'}, Y_{i,j})$
29. $\pi_{i,j,j'} = cost^{current} + cost(Y_{i-1,j'})$
30. **End for**
31. Let $cost(Y_i)$ equals to the minimum value in $\pi_{i,j}$ and records the relevant deployment node and link.
32. **End for**
33. **End for**
34. The minimum value recorded in $cost(Y_{i,j})$ is the required minimum bandwidth occupation deployment results
35. Update the network resource status
36. **End if**

C. Complexity Analysis

We simply analyze the computational complexity in this section. Let the maximum number of VNFs in the SFC request is m, and the maximum number of servers is n in the substrate network. In our algorithm, the computing complexity of determining the set of servers that meet the deployment requirements for each VNF_i^r is $o(mn)$. Then, the computational complexity of the primary link deployment process with the Viterbi algorithm as its core is $o(mn^2)$. The maximum computational complexity for the heterogeneous backups based on the Viterbi algorithm is also $o(mn^2)$. Therefore, the total computational complexity of the algorithm is $o(mn^2)$.

IV. Experimental Simulation

A. Simulation Setup

We use the DC.K8 data center network topology provided by [13], which contains 80 switch nodes and 256 links. The link bandwidth capacity selected from {10, 20, 30, 40}. And 32 of all the switch nodes are deployed with server nodes. There are six types of server nodes, and they are subject to the uniform distribution of [1, 6]. The computational resources for each sever node are randomly selected from {16, 32, 48, 64}. There are six types of VNFs, and each VNF has two heterogeneous functionally identical VNFs. The resource requirement coefficients for each VNF are shown in Table II. The types of VNF that each server node can load are randomly selected 4 from the 6 type VNF. The size of the traffic that each SFC request needs to process is subject to the uniform distribution of [1, 4]. The request includes 4 kinds of VNFs, which are randomly selected from the 6 types of VNFs. The arrival of the request obeys a Poisson process with a parameter of 0.05 and the life cycle obeys an exponential distribution with a mean of 1000 units of time. We implemented the proposed algorithms in Python. All these algorithms were executed on a computer equipped with an Intel(R) Core(TM) i5 CPU 2.60GHz processor with 2 cores, and 8GB of RAM. And we used the Matlab to analyze the experimental results.

TABLE II. Computing Resources Coefficient

VNF	Computing resource required/ Unit bandwidth
Firewall_x	1/ Unit bandwidth
Proxy_x	1/ Unit bandwidth
Nat_x	2/ Unit bandwidth
IDS_x	3/ Unit bandwidth
DPI_x	3/ Unit bandwidth
Encryption_x	4/ Unit bandwidth

In order to evaluate the feasibility and validity of the algorithm, this experiment applies the Request acceptance rate, the Average bandwidth consumption per request, and the Anti-attack ability as evaluation indicators. Three algorithms are involved in the comparison, including No backup algorithm [13], Backup algorithm without considering heterogeneity, and Heterogeneous backup algorithm proposed in this paper.

B. Performance Analysis

So as to simplify the description, we use No-Backup (No backup) to indicate the algorithm that does not back up any VNF, let Half-Backup (half backup) to represent the algorithm that backs up half of the key VNFs without consider heterogeneity, use Half- HetBackup (half backup with heterogeneous nodes) to denote the algorithm that backs up half of the key VNFs with heterogeneous node, and let All-HetBackup (all backup with heterogeneous nodes) to represent the algorithm that backs up all VNFs with heterogeneous node.

1) Request acceptance rate

Figure 5 (a) shows the request acceptance rate for the four algorithms at a certain request strength. Performing redundant backups will consume more physical resources. Therefore, when the algorithm performs backups, the request reception rate will decrease significantly. In the case of backing up 50% of VNFs, compared to b, a due to considering the heterogeneity of the backup node will discard some available server nodes, that will lead to a certain degree of request acceptance rate decline. According to the experimental data, the average decrease was 4.81% after stabilization.

2) Average bandwidth consumption per request

Figure 5(b) shows the average bandwidth consumption per request for the four algorithms. When doing VNF backup, the algorithm needs to build links between backup node and related VNFs in the primary link, and links between backup

VNFs, resulting in the need to deploy more links. When performing heterogeneous backup, taking into account the heterogeneity of the backup server nodes, some near but homogeneous server nodes are discarded, resulting in a certain increase in the length of the link. According to the experimental data, the average increase of bandwidth is 5.23 unit.

3) Anti-attack ability

In this part, we compare the anti-attack performance of SFC in four situations. We have simply modeled the attack process. Assume that the time required for an attacker to successfully break a VNF on a type of server node obeys an exponential distribution with a mean of 1,000. Attackers will continue to attack VNFs in the same SFC until the SFC service is interrupted. We simulated and recorded the time taken by the attacker to interrupt the SFC service in four cases and repeated 100 experiments. We performed a statistical analysis of the attack time, plotted the cumulative distribution function of attack time as shown in Figure 6(a), and the average attack time as shown in Figure 6(b). As can be seen from Figure 6(a), the backup strategy can significantly increase the attacker's attack. As shown in Figure 6(b), the attacker's average attack time is 2059 units in the case of Half-HetBackup, and 1819 units in the case of Half-Backup. The Half-HetBackup algorithm increases the attack time cost by 13.2% compared to the method without considering the heterogeneity.

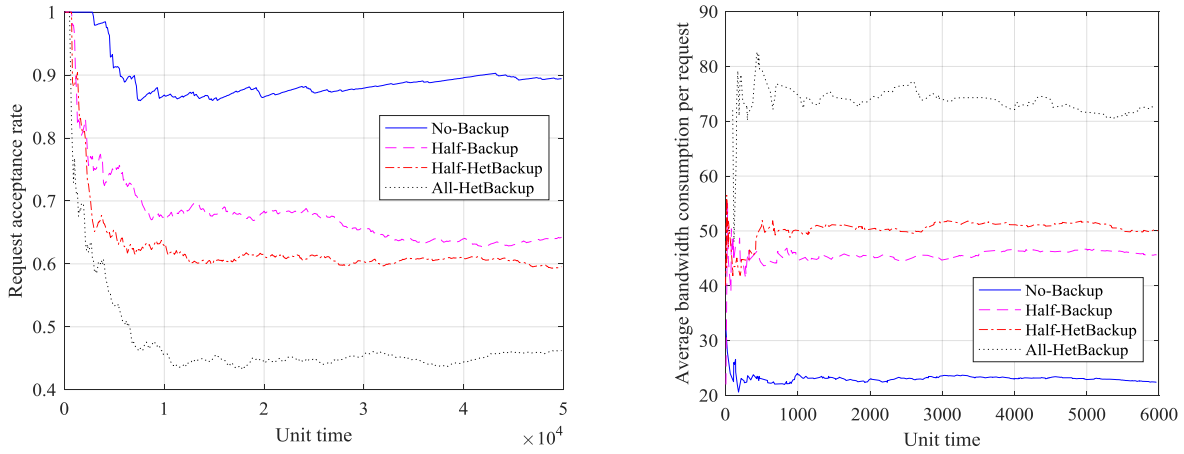

Figure 5. Request Acceptance Rate and bandwidth consumption: (a) Request acceptance rate. (b) Average bandwidth consumption per request.

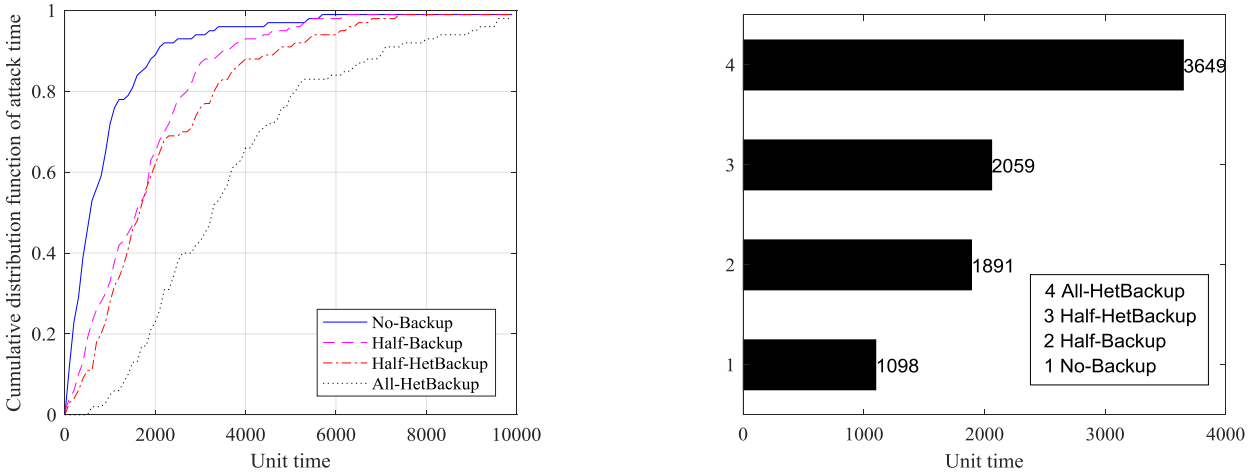

Figure 6. Anti-attack ability: (a) Cumulative distribution function of attack time. (b) Average attack time.

V. CONCLUSION

This paper firstly analysis the existing redundancy backup deployment methods of the service function chain which do not consider the node homogeneity problem. And then proposes a heterogeneous backup deployment scheme. Moreover, the objective function is to minimize the link bandwidth consumption. In this paper, a SFC deployment model with heterogeneous backups was constructed and a corresponding algorithm was designed. The experimental results show that considering the heterogeneity of the backup node and the original node during backup can significantly increase the attacker's attack time and further improve the reliability of SFC. In the future work, we will further study

the deployment of SFC. We plan to use backup sharing strategy to further improve the resource utilization and the request acceptance rate. Moreover, we plan to solve the problem of privacy information leaked that SFC faces due to the sharing of underlying physical resources.

ACKNOWLEDGMENT

This research has been partly supported by the National Key Research and Development Project of China under Grant No. 2016YFB0801200, the National Cyberspace Security Special Project of China under Grant No. 2017YFB0803204, the National Natural Science Foundation of China under Grant No. 61521003, and the Ministry of Public Security Science and Technology General Project under Grant No. 2017JSYJC08.

REFERENCES

[1] Medhat A M, Taleb T, Elmangoush A, et al. Service Function Chaining in Next Generation Networks: State of the Art and Research Challenges[J]. IEEE Communications Magazine, 2017, 55(2):216-223.

[2] Mijumbi R, Serrat J, Gorricho J L, et al. Network Function Virtualization: State-of-the-art and Research Challenges[J]. IEEE Communications Surveys & Tutorials, 2017, 18(1):236-262.

[3] NFV ISG, "Network Functions Virtualisation - An Introduction, Benefits, Enablers, Challenges & Call for Action," ETSI, Tech. Rep., 2012. [Online]. Available: http://portal.etsi.org/NFV/NFVn Whiten Paper.pdf

[4] John Donovan. How Do You Keep Pace With a 100,000 Percent Increase in Wireless Data Traffic? Software. http://about.att.com/innovationblog/3215howdoyoukeeppace

[5] Cotroneo D, Simone L D, Iannillo A K, et al. Network Function Virtualization: Challenges and Directions for Reliability Assurance[C]// IEEE International Symposium on Software Reliability Engineering Workshops. IEEE Computer Society, 2014:37-42.

[6] Medhat A M, Carella G A, Pauls M, et al. Resilient orchestration of Service Functions Chains in a NFV environment[C]// Network Function Virtualization and Software Defined Networks. IEEE, 2017:7-12.

[7] Hmaity A, Savi M, Musumeci F, et al. Virtual Network Function placement for resilient Service Chain provisioning[C]// International Workshop on Resilient Networks Design and Modeling. IEEE, 2016.

[8] Herker S, An X, Kiess W, et al. Data-Center Architecture Impacts on Virtualized Network Functions Service Chain Embedding with High Availability Requirements[C]// IEEE GLOBECOM Workshops. IEEE, 2015:1-7.

[9] Casazza M, Fouilhoux P, Bouet M, et al. Securing Virtual Network Function Placement with High Availability Guarantees[J]. 2017:1-9.

[10] Beck M T, Botero J F, Kai S. Resilient allocation of service Function chains[C]// Network Function Virtualization and Software Defined Networks. IEEE, 2017:1-6.

[11] Ding W, Yu H, Luo S. Enhancing the reliability of services in NFV with the cost-efficient redundancy scheme[C]// IEEE International Conference on Communications. IEEE, 2017:1-6.

[12] Carpio F, Jukan A. Improving Reliability of Service Function Chains with Combined VNF Migrations and Replications[J]. 2017.

[13] Bari F, Chowdhury S R, Ahmed R, et al. Orchestrating Virtualized Network Functions[J]. IEEE Transactions on Network & Service Management, 2016, PP(99):1-1.