IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 17, 2019, accepted July 4, 2019, date of publication July 9, 2019, date of current version July 26, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2927490

Survivable Virtual Network Link Shared Protection Method Based on Maximum Spanning Tree

XIAOYANG HAN, XIANGRU MENG, QIAOYAN KANG, AND YUZE SU
College of Information and Navigation, Air Force Engineering University, Xi'an 710077, China

Corresponding author: Xiaoyang Han (dodogg189888@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61401499 and Grant 61871313.

ABSTRACT Network virtualization allows multiple independent and mutual non-interfering virtual networks (VNs) to run simultaneously on a shared substrate network (SN) providing a promising method for overcoming the network ossification. Substrate infrastructure failures occur frequently, most of which are substrate link failures. This has created a research hotspot in the field of survivable VN. In this paper, we add to the research by analyzing the substrate link failures and proposing a survivable VN link shared protection method based on the maximum spanning tree. First, an integer linear programming formulation for the survivable VN link shared protection is established to minimize the backup link bandwidth ratio. Second, an efficient VN mapping algorithm based on optimal matching is designed to make full use of the limited substrate resources. This mapping algorithm that fully considers the topological connections between nodes in the node mapping phase improves the acceptance ratio and the long-term average revenue to cost ratio. Finally, a maximum spanning tree link protection algorithm is proposed, and a reasonable sharing mechanism of backup link resources among different VNs is designed to ensure the VN survivability and reduce the SN resource consumption. The experimental results show that our method not only guarantees the VN survivability but also reduces SN resource consumption.

INDEX TERMS Survivable virtual network, substrate link failure, link shared protection, optimal matching, maximum spanning tree, sharing mechanism.

I. INTRODUCTION

With the rapid growth of the Internet, the problem of network ossification is worsening. It has become a key restrictor of innovation and further development of Internet architecture [1]–[3]. Network virtualization which allows multiple independent, mutual non-interfering VNs to run simultaneously on a shared SN and VNs run different protocols and applications can effectively solve the network ossification problem. Hence, it has received extensive attention from the researchers [4], [5]. VN mapping whose main task is to map the virtual nodes/links onto the substrate nodes/links on the premise of satisfying the VN mapping constraints is the key method of realizing network virtualization. It has been proven that the VN mapping problem is NP-hard. To improve the acceptance ratio, utilize the SN resources efficiently, and reduce the mapping costs, an effective VN mapping algorithm must be designed to deal with the dynamic VN mapping requests. Many researchers have proposed innovative mapping algorithms, for this purpose [6]–[10].

Previous studies on VN mapping aimed to optimize the utilization of SN resources without considering the VN survivability when substrate infrastructure failure occurred. VN failure ocurrs and the continuity of network services is affected when the hosted substrate infrastructure fails. The network service interruption caused by the substrate infrastructure failure can create great losses for users and force the network service providers to pay high compensation. Therefore, improving the VN survivability has become the focus of network virtualization research. Owing to the limitations of the SN resources, the VN survivability should be guaranteed while the substrate network resource consumption should be reduced as much as possible, so the problem becomes more complicated [11], [12].

VN survivability refers to the ability of VN to accomplish the key tasks on time when the substrate infrastructure

The associate editor coordinating the review of this manuscript and approving it for publication was Ting Wang.

failures occur. It focuses on continuity of network services and failure resilience to ensure VN fast recovery and to minimize the effects on the network services. The methods used to deal with the failure of substrate infrastructure can be divided into two categories: recovery and protection [13], [14]. The recovery mechanism recovers VN by remapping the affected virtual nodes/links and does not need to reserve redundant resources. Resource consumption is thus lower. However, the recovery mechanism often needs a recovery time much larger than the 50 ms, and the network delay which has a great effect on the network service is inevitable. Limited by the remaining available SN resources, the acceptance ratio of node/link remapping cannot be guaranteed. These facts affect the application of the recovery mechanism [15], [16].

The principle of the protection mechanism is reserving redundant resources for virtual nodes/links in the mapping process. It enables quick switching of the failed VN nodes/ links to the backup nodes/paths when the substrate infrastructure failure occurs. The goal is to recover the network within 50 ms, so that the continuity of the network services can be guaranteed. The protection mechanism is highly favored in the research of VN survivability although it will cause a large consumption of SN resources in the backup mapping process. Thus, reducing resource consumption while ensuring VN survivability has become an urgent problem [17]–[19].

Most incidences of the substrate infrastructure failure are caused by the substrate link failures, so it is urgent to strengthen the research of VN survivability to deal with the failure of substrate link. The link protection is a hot issue with respect to survivable VN, because it can recover the failed virtual link quickly, decrease the effect to the network services and reduce the resource consumption of SN by designing an efficient link protection method. Many innovative methods of survivable virtual link protection proposed by researchers have improved the VN survivability when substrate link failure occurs. They have reduced the consumption of SN resources, and have promoted the survivable VN research [20]–[27]. According to the current knowledge, link protection methods can be divided into dedicated backup protection and shared backup protection. Dedicated backup protection reserves dedicated resources for each virtual link, and each backup link resource only used for its primary link to recover the network quickly. The network can be quickly recovered after the failure because of the sufficient reserved resources. Therefore, the dedicated backup protection is an effective method for dealing with the link failure. A link backup protection method was proposed in [20], introducing a dedicated protection scheme to recover the link failure. Although the VN survivability could be greatly improved, the SN resource consumption was large.

Some researchers have proposed shared backup protection to solve the large resource consumption problem of dedicated backup. Shared backup protection also reserves redundant link resources, but they can be shared among different VNs. The purpose of shared backup protection is to minimize the redundant resources while maintaining VN survivability. Although the complexity of the shared backup protection method is higher, it guarantees the VN survivability, obviously improve the resource utilization rate and reduces the consumption of the SN resources. A VN survivable method which supports path splitting to deal with the substrate link failure was proposed in [21]. This method effectively dealt with single substrate link failures and reduced reserved redundant resources. However, path splitting increased the incidence of virtual link failure. A virtual link sharing backup method that fully backed-up the virtual links and the backup path could be shared was proposed to reduce the consumption of the SN resources in [22]. However, backing up all virtual links caused a waste of SN resources. Optimal topology backup was considered in [23], where the resource consumption was reduced by optimizing and simplifying the VN backup topology. However, this method only dealt with the single substrate link failure. The backup resource utilization still needs to be improved.

Several problems remain to be solved, which mainly include: (i) the backup topology can be further simplified and optimized, and the utilization of backup link resources can be improved. (ii) Most of the prior approaches generally do not consider the occurrence of multi-substrate link failures. To solve these problems, we propose a survivable virtual network link shared protection method based on maximum spanning tree (SVNLSPM-MST) in this paper. First, the new survivable VN link shared protection method is provisioned, optimally via integer linear programming (ILP) on the basis of analyzing the substrate link failures. This model is built to protect the VN links and to reasonably share the backup link resources. Second, we calculate the maximum spanning tree (MST) of VNs and propose an efficient VN mapping algorithm. This algorithm gives full consideration to the topological connections in the node mapping phase, and maps the MST link and the non-maximum spanning tree (NMST) link separately. Finally, we design an MST link protection algorithm (LSP-MST) that offers direct backup protection to the MST link and gives indirect backup protection to the NMST link by sharing the backup resources with the MST link. This helps cope with the possibility of substrate link failure. A reasonable sharing mechanism is included in the LSP-MST to improve the utilization rate of substrate link resources on the premise of ensuring VN survivability.

The main contributions of this paper can be summarized as follows.

(i) We provide an ILP formulation for survivable link shared protection and propose a VN mapping algorithm which includes virtual node mapping based on the optimal matching (OM-VNM) and virtual link mapping based on improved k-shortest path (I-k-SPVLM). This algorithm fully considers the topological connections between nodes during the node mapping phase, ensuring the mapping performance and reducing the SN resource consumption.

(ii) The LSP-MST is proposed, further optimizing the backup topology and making the backup resource sharing more reasonable. Thus, it guarantees the survivability of VNs and reduces the SN resource consumption as much as possible. When a single substrate link failure occurs, the affected virtual link can be immediately switched to the backup link to quickly recover the VN and the effect to the network service can be reduced as much as possible. When a multi-substrate link failure occurs, it can maintain the network connectivity through service degradation and reduce the effects to the network services. A backup link resource sharing mechanism is also designed in the LSP-MST to ensure that the backup link resources of different VNs can be reasonably shared. Thus the utilization rate of SN resources can be improved on the premise of ensuring the rapid recovery of affected virtual links when the substrate link failures occur.

The rest of this paper is organized as follows. In section II, we present related work. In section III, we present the problem statement. In section IV, the evaluation indicators are presented. In section V, we propose the ILP of the survivable virtual link protection. The SVNLSPM-MST is given, and its details are shown in section VI. In section VII, we validate and evaluate the proposed method with extensive simulations and experiments. We conclude this paper in section VIII.

II. RELATED WORK

Reserving redundant resources to protect the virtual links is a promising method to deal with the substrate link failures. It can quickly switch the affected VN links to the backup paths when the substrate infrastructure fails, keeping the recovery time within the allowable limits. Thus, the continuity of the network services can be guaranteed, and the VN survivability can be improved. The objective of virtual link protection is to maximize the VN survivability while minimizing the total resources required for recovery.

With respect to the issue of virtual link protection, many scholars have proposed solutions. They can be generally divided into two categories: dedicated backup protection and shared backup protection. The dedicated backup protection is an effective way to deal with the link failure. To solve the survivable VN link protection problem, a proactive and hybrid heuristic policy based on a fast rerouting strategy was proposed, providing dedicated backup to all physical links in [24]. This method improved the VN survivability when the substrate link failure occurred, it also caused great resource consumption. To cope with the high network resource consumption in dedicated protection, many shared protection methods are proposed. Backup-sharing can dramatically improve the backup resource utilization while ensuring the VN survivability. A new link protection method was proposed, introducing a novel prognostic redesign technique that augmented the VN topology and promoted the backup resource sharing prior to the VN mapping phase in [25]. A VN mapping algorithm for node reliability perception and link shared protection was proposed to improve the VN survivability, making a trade-off between survivability and resource consumption [26]. Although these methods consider the sharing of the backup link resources in the VN mapping phase, the backup topology was equal to the virtual network, and link resource consumption was still large.

Most of the prior approaches for survivable VN design aim at single substrate link failure scenario, which are not applicable to the more challenging multi-substrate link failure scenario. To overcome this problem, a new method was proposed in [27], which introduced the spanning tree into the link protection to deal with multi-substrate link failures. This was a novel method that dealt with the multi-substrate link failures to an extent. However, owing to limited redundant resources, some virtual links affected by the substrate link failure could not be recovered.

Based on current link protection methods, some researchers have innovatively applied P-cycle and network coding technologies to the virtual link protection, obtaining better results [28]–[33]. Network coding technology was applied in [31] to improve the survivability of VN. However, network coding will occupied node resources and increased node loads in the application process. A link protection method based on network coding and P-circle link protection was creatively proposed in [32], augmenting the virtual topology prior to the VN mapping to improve the VN survivability and reduce the network resource consumption. However, the method was relatively complicated and could not protect all VNs.

These innovative virtual link protection methods promoted the development of survivable VN, but there are still some problems need to be solved which mainly include: (i) the backup topology was equal to the VN in most of the prior approaches, and there remains some room for improvement in utilization of SN resources. (ii) Most of the prior approaches emphasized dealing with single substrate link failures, and did not consider the occurrence of multi-substrate link failures. (iii) The designs of some link protection methods were also complicated.

To effectively address these shortcomings, we propose the SVNLSPM-MST in this paper. By further optimizing the backup topology, our method fully deals with single substrate link failures and reacts to multi-substrate link failures to an extent. To the best of our knowledge, this is the first attempt to introduce the MST for virtual link protection and to use the optimal matching to solve the survivable VN mapping problem. Thus, resource consumption is reduced and the VN survivability is improved.

III. PROBLEM STATEMENT

This paper studies the survivability of VN when substrate link failures occur and designs an SVNLSPM-MST, which can fully deal with single substrate link failures and react to multi-substrate link failures to some extent with lower substrate link resource consumption.

A. ANALYSIS OF THE SUBSTRATE LINK FAILURE

1) SINGLE SUBSTRATE LINK FAILURE I

As shown in Fig. 1, the virtual link (b, c) of VN1 and the virtual link (d, f) of VN2 are mapped on the substrate

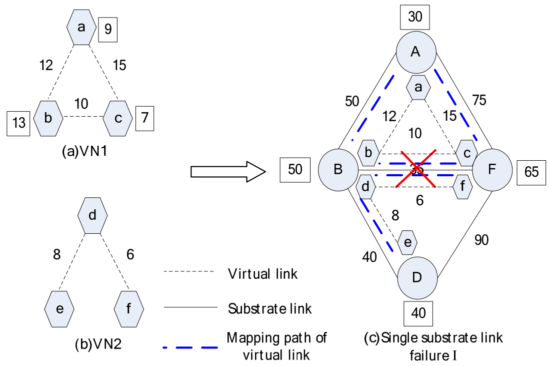


FIGURE 1. An example of single substrate link failure I.

link (B, F) simultaneously. When the substrate link (B, F) fails, the virtual links (b, c) and (d, f), hosted by substrate link (B, F) fail at the same time.

2) SINGLE SUBSTRATE LINK FAILURE II

As shown in Fig. 2, the virtual links (a, b) and (b, c) of VN are respectively mapped to the SN paths (A, B) (B, A,F), and their mapping SN paths overlap. When the substrate link (A, B) fails, the virtual links (a, b) and (b, c) fail at the same time.

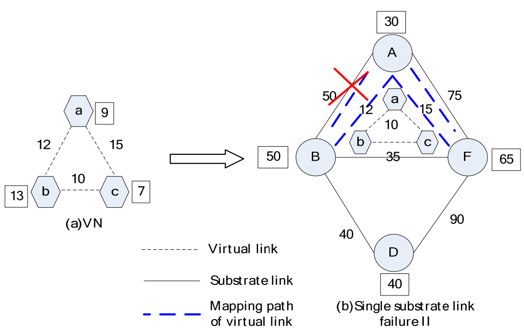


FIGURE 2. An example of single substrate link failure II.

3) MULTI-SUBSTRATE LINK FAILURE

As shown in Fig. 3, when the substrate links (A, B), (B, C), and (A, C) fail at the same time, it leads to the failure of the virtual links (a, b), (b, c), and (a, c) which are respectively hosted by substrate links (A, B), (B, C), and (A, C), causing multiple link failures in a VN.

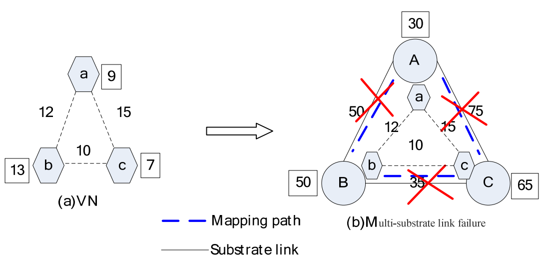


FIGURE 3. An example of multi-substrate link failure.

B. LINK FAILURE RECOVERY

If all virtual links are protected by backup resources, when a link failure occurs, the effect to the network service can be reduced significantly by quickly switching to the backup link. It is rare under realistic situations for multiple links to fail at the same time. Thus, dedicated backup for all virtual links would cause a great waste of resources. Therefore, a trade off must be made between ensuring network survivability and reducing resource consumption, that means reducing the resource consumption and ensuring the network survivability. For this purpose, we adopt the MST link protection method in the virtual network link protection. The spanning tree is a minimum connected graph that includes all vertices of the graph. The MST is a spanning tree whose sum of edge weights is the largest, and it can be obtained by using the improved Kruskal algorithm [34]. As shown in Fig.4, Fig. 4(b) is the MST of Fig. 4(a). The MST link is directly protected by reserving backup substrate link resources, and the NMST link is indirectly protected by sharing the backup link resources of the MST link.

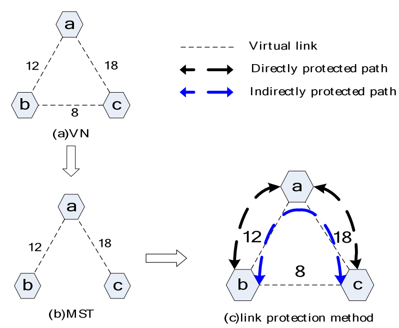


FIGURE 4. The MST link protection method.

Owing to the connectivity of the spanning tree, when a single substrate link failure occurs, the virtual link hosted by the failed substrate link can be recovered in a timely manner. When a multi-substrate link failure occurs, we can ensure the basic connectivity of the VN nodes via service degradation. As shown in Fig. 5, we divide the virtual link into MST link (a, b), (a, c), and NMST link (b, c). The virtual link mapping scheme is {(a, b)→(B, D), (a, c)→(B, F), (b, c)→(D, E,F)}. The MST link backup protection scheme is {(a, b)→(B, C, D), (a, c)→(B, A, F)}. The NMST link is protected indirectly

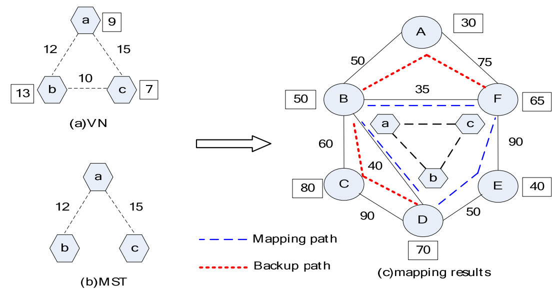


FIGURE 5. An example of virtual link backup protection.

by sharing the backup link resources with the MST link: {(b, c)→(D, C,B, A,F)}. If the mapping paths of the NMST link (b, c) and the MST links (a, b) and (a, c) overlap, when substrate link (B, D) or (B, F) fails, the virtual link (b, c) cannot immediately be recovered. Thus, in the VN mapping process, we should avoid mapping the MST links and the NMST links to the same SN path or their mapping SN paths overlap.

The link failure recovery scheme for the single substrate link failure I scenario is as follows. When substrate link (B, D) or (B, F) failure occurs, the VN MST links (a, b) and (a, c), which are hosted by the failure link, will fail. The failed virtual link can be quickly switched to the backup path (B, C,D) or (A, B,F). The failed VN can be fast recovered, so the normal operation of the VN and the continuity of network service can be guaranteed. The link failure recovery scheme for the single substrate link failure II scenario is as follows. When the substrate link (D, E,F) fails, the NMST link (b, c) of the VN, which is hosted by the failure substrate link, will fail. Via backup resource sharing, the failed virtual link can be quickly switched to the indirect protection path (D, C, B, A, F) to recover the VN quickly. For the multi-substrate link failure, when the substrate links (B, D), (B, F) and (D, E, F) fail at the same time, only the MST links (a, b) and (a, c) of the VN can be quickly switched to the direct backup path (B, C, D) and (B, A, F) to partially restore the VN and ensure basic connectivity of the VN nodes.

C. BACKUP LINK RESOURCE SHARING

Because the probability of multi-substrate links failing simultaneously is low, dedicated backup to all virtual links will cause a waste of resources. Thus the backup link resource shared by different VNs should be considered. When a single substrate link failure occurs, if the backup link resource sharing is not reasonable, some of the failed links cannot be able to recover quickly due to the insufficient resources.

As shown in Fig. 6, The virtual link mapping scheme of VN1 is {(a, b)→(A, B)}, and its backup protection scheme is {(a, b)→(A, F,B)}. The virtual link mapping scheme of VN2 is {(c, d)→(B, D), (c, e)→(B, A,F)}, and its backup protection scheme is {(c, d)→(B, F,D), (c, e)→(B, F)}. The backup mapping paths of virtual links (c, e), (a, b), and (c, d) overlap. If virtual links (c, e), (a, b), and (c, d) share the same virtual link backup resources, the failure of the SN path (A, B) will cause the virtual links (a, b) and (c, d) to fail at the same time. We cannot recover the virtual links (a, b) and (c, d) simultaneously because of the limited backup resources. To solve this problem, we should ensure that the virtual links of different VNs mapped to the same SN path cannot share the same backup resources when their backup link is mapped to the same SN path.

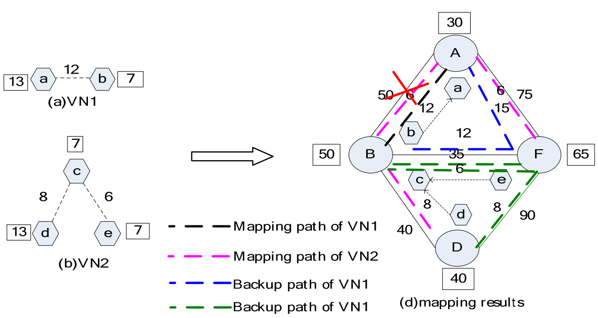


FIGURE 6. An example of backup link sharing.

IV. NETWORK MODEL AND EVLUATION INDICATORS

A. NETWORK MODEL

The SN is modeled as a weighted undirected graph $\boldsymbol{G^s} = (N^s, E^s)$. The VN can be modeled as a weighted undirected graph $\boldsymbol{G^v} = (N^v, E_P^v)$, and the VN backup topology can also be modeled as a weighted undirected graph, $\boldsymbol{G_B^v} = (N^v, E_B^v)$.We define the corresponding symbols in Table 1.

TABLE 1. Description of symbols.

Symbol	Description	Symbol	Description
G^s	SN	G^v	VN
n^s	substrate node	n^v	virtual node
N^s	substrate node set	N^v	virtual node set
e^s	substrate link	$e_P{}^v$	virtual primary link
-	-	$e_B{}^v$	virtual backup link
E^s	substrate link set	$E_P{}^v$	virtual primary link set
-	-	$E_B{}^v$	virtual backup link set
$cpu(n^s)$	CPU resources of n^s	$cpu(n^v)$	required CPU resources of n^v
$bw(e^s)$	available bandwidth resources of e_s	$bw(e^v)$	required bandwidth resources of e^v
$\|N^s\|$	number of substrate nodes	$\|N^v\|$	number of virtual nodes
$\|E^s\|$	number of substrate links	$\|E^v\|$	number of virtual links

B. EVLUATION INDICATORS

1) ACCEPTANCE RATIO

The acceptance ratio is a key indicators describing the performance of VN mapping and the overall resource utilization. It can be defined as follows.

$$r = \lim_{T\to\infty} \frac{\sum_{t=0}^{T} VN_{suc}(t)}{\sum_{t=0}^{T} VN(t) + \delta} \tag{1}$$

Here, $|VN(t)|$ is the number of VN requests at time t, $|VN_{suc}(t)|$ is the number of VNs that are successfully protected by link backup at time t, and δ is a constant which is infinitely close to 0.

2) LONG-TERM AVERAGE REVENUE TO COST RATIO

For VN request $\mathbf{G_v} = (N_v, E^v, CB(n_v))$, we denote the revenue $R(G_v, t)$ and the cost $C(G_v, t)$ as follows.

$$R(\mathbf{G_v}, t) = \alpha \sum_{n^v \in N^v} cpu(n^v) + \sum_{e^v \in E^v} bw(e^v) \tag{2}$$

$$C(\mathbf{G_v}, t) = \beta \sum_{n^v \in N^v} cpu(n^v) + \sum_{e^v \in E^v} hops(M(e^v)) \cdot bw(e^v) \tag{3}$$

In which, α and β are weighting coefficients used to balance the CPU and bandwidth resources, respectively. In this paper, we assume $\alpha = \beta = 1$, indicating that the importance of CPU and bandwidth is similar. $hops(M(e^v))$ is the hop count of the substrate path corresponding to the virtual link e^v. Usually, the long-term average revenue to cost ratio is used to represent the performance of the VN mapping algorithm under a steady state. It can be defined as follows.

$$R/C = \lim_{T\to\infty} \frac{\sum_{t=0}^{T} \sum_{\mathbf{G_v} \in VN_{\mathbf{map}}(t)} R(\mathbf{G_v}, t)}{\sum_{t=0}^{T} \sum_{\mathbf{G_v} \in VN_{\mathbf{map}}(t)} C(\mathbf{G_v}, t)} \tag{4}$$

3) BACKUP LINK BANDWIDTH RATIO

The backup link bandwidth ratio denotes the utilization of link backup resources, which can be defined as follows.

$$BLBR = \lim_{T\to\infty} \frac{\sum_{t=0}^{T} \sum_{e_B^v \in E_B^v} hops(M(e_B^v)) \cdot bw(e_B^v)}{\sum_{t=0}^{T} \sum_{e_P^v \in E_P^v} hops(M(e_P^v)) \cdot bw(e_P^v)} \tag{5}$$

Here, $hops(M(e_B^v))$ and $hops(M(e_P^v))$ are used to denote the hop counts of the backup virtual link e_B^v and the primary virtual link e_P^v, respectively.

V. ILP OF SURVIVABLE VN LINK SHARED PROTECTION

Owing to the limited substrate resources, improving the utilization of SN resources has become one of the important directions in the research of VN survivability. In this section, we formulate the ILP of survivable VN link shared protection. The objective function and constraints can be expressed as follows.

A. OBJECTIVE FUNCTION

In this paper, our object is to obtain the minimum backup link bandwidth ratio on the premise of ensuring the VN survivability.

$$\min \left\{ \lim_{T\to\infty} \frac{\sum_{t=0}^{T} \sum_{e_B^v \in E_B^v} hops(M(e_B^v)) \cdot bw(e_B^v)}{\sum_{t=0}^{T} \sum_{e_P^v \in E_P^v} hops(M(e_P^v)) \cdot bw(e_P^v)} \right\} \tag{6}$$

B. CONSTRAINTS

$$\forall n_i^v \in N^v, \quad \forall n_j^s \in N^s$$

$$x_s^v(i,j) = \begin{cases} 1, & if\ n_i^v\ is\ mapped\ onto\ n_j^s \\ 0 & otherwise \end{cases} \tag{7}$$

$$\forall e_{im}^v \in E^v, \quad \forall e_{jl}^s \in E^s$$

$$y_{jl}^{im}(e_{im}^v, e_{jl}^s) = \begin{cases} 1, & if\ e_{im}^v\ is\ mapped\ onto\ e_{jl}^s \\ 0 & otherwise \end{cases} \tag{8}$$

Constraint (7) indicates that, if virtual node n_i^v is mapped to substrate node n_j^s, $x_s^v(i,j) = 1$. Otherwise, $x_s^v(i,j) = 0$. Constraint (8) indicates that, if virtual link e_{im}^v is mapped to the substrate path e_{jl}^s, $y_{jl}^{im} = 1$. Otherwise, $y_{jl}^{im} = 0$.

$$\forall e_{im}^v \in E_{MCST}^v, \quad \forall e_{mp}^v \in E_{NMCST}^v, \ \forall e_{jl}^s \in E^s, \ \forall e_{jl}^s \in e_{uq}^s$$

$$z(e_{im}^v, e_{jl}^s) = \begin{cases} 1, & if\ e_{im}^v\ is\ protected\ by\ e_{jl}^s\ directly \\ 0 & otherwise \end{cases}$$

$$z(e_{mp}^v, e_{uq}^s) = \begin{cases} 1, & if\ e_{mp}^v\ is\ protected\ by\ e_{uq}^s\ indirectly \\ 0 & otherwise \end{cases} \tag{9}$$

Constraint (9) indicates that, if the MST link of VN e_{im}^v is directly protected by the substrate path e_{jl}^s, $z(e_{im}^v, e_{jl}^s) = 1$, otherwise, $z(e_{im}^v, e_{jl}^s) = 0$, and if the NMST link of VN e_{mp}^v is indirectly protected by the substrate path e_{uq}^s, $z(e_{mp}^v e_{uq}^s) = 1$. Otherwise, $z(e_{mp}^v e_{uq}^s) = 0$.

$$if\ z(e_{im}^v, e_{jl}^s) = 1$$

$$z(e_{im}^v, e_{jl}^s) + z(e_{mp}^v, e_{lk}^s) \le 1 \tag{10}$$

$$\forall n_i^v \in N^v, \quad \forall n_j^s \in N^s$$

$$x_s^v.cpu(n_i^v) \le cpu(n_j^s) \tag{11}$$

$$\forall n_j^s \in N^s, \quad \sum_{n_i^v \in N^v} x_s^v \le 1 \tag{12}$$

Constraint (10) denotes that the MST link which is protected directly, and the NMST link which is protected indirectly, cannot be mapped to the same substrate path or mapping path overlaps. Constraint (11) indicates that the remaining CPU resources of the substrate node should not be less than the CPU resource demand of the virtual node, which is hosted by the substrate node. Constraint (12) indicates that each substrate node can host at most one virtual node in the same VN mapping request.

$$\forall n_i^v \in N^v, \quad \sum_{n_j^s \in N^s} x_s^v = 1 \tag{13}$$

$$\forall e_{im}^v \in E^v, \quad \forall e_{jl}^s \in E^s$$

$$\sum_{e^{ij} \in E^s} y_{jl}^{im}(e_{im}^v, e_{jl}^s).b(e_{im}^v) \le b(e_{jl}^s) \tag{14}$$

$$\forall n_i^v \in N^v, \quad \forall n_j^v \in N^s, \ \forall e_{im}^v \in E^v, \forall e_{jl}^s \in E^s$$

$$\sum_{e_{jl}^s \in E^s} y_{jl}^{im}(e_{im}^v, e_{jl}^s) - \sum_{e_{lj}^s \in E^s} y_{lj}^{im}(e_{im}^v, e_{lj}^s)$$

$$= x_s^v(i,j) - x_s^v(m,l) \tag{15}$$

Constraint (13) indicates that each virtual node can only be mapped to one substrate node. Constraint (14) indicates that the link bandwidth requirement in the VN request must be less than or equal to the remaining bandwidth of the substrate link. Constraint (15) indicates that the virtual link should be mapped to the acyclic path of the SN.

$$\forall n_i^v \in N^v, \quad \forall n_j^s \in N^s$$

$$x_s^v(i,j) \in \{0,1\} \tag{16}$$

$$\forall e_{im}^v \in E^v, \quad \forall e_{jl}^s \in E^s$$

$$y_{jl}^{im}(e_{im}^v, e_{jl}^s) \in \{0,1\} \tag{17}$$

$$\delta(e_{im}^v, e_{jlb}^s) = \begin{cases} 1 & if\ e_{im}^v\ isprotected\ by\ e_{jlb}^s \\ 0 & otherwise \end{cases} \tag{18}$$

$$if\ \delta(e_{im}^v, e_{jlb}^s) = 1$$

$$y_{jl}^{im}(e_{im}^v, e_{jl}^s) + y_{jl}^{im}(e_{im}^v, e_{jlb}^s) \le 1 \tag{19}$$

Constraints (16) and (17) represent the limitations of variables ranges. Constraints (18) and (19) indicate that the backup mapping path and the original mapping path of the MST link cannot overlap.

$$\theta(e_{jl1}^s, e_{jl2}^s) = \begin{cases} 0 & if\ e_{im1}^v, \quad e_{im2}^v\ is\ embeded\ to\ e_{jl1}^s \\ & \&\ e_{im1b}^v, \quad e_{im2b}^v\ is\ embeded\ to\ e_{jl2}^s \\ 1 & otherwise \end{cases} \tag{20}$$

$$if\ \theta(e_{jl1}^s, e_{jl2}^s) = 0$$

$$y_{jl}^{im}(e_{im1}^v, e_{jl1}^s) + y_{jl}^{im}(e_{im2}^v, e_{jl2}^s) \le 1 \tag{21}$$

$$\partial(e_{lm\ new}^v) = \begin{cases} 0, & val = val + 1; \quad ife_{lm\ new}^v \le e_{jlb}^s \\ 1, & val = 1; \quad if\ e_{lm\ new}^v > e_{jlb}^s \end{cases} \tag{22}$$

Constraints (20) and (21) indicate that, if the virtual links of different VN mapped to the same path or if the mapping path overlaps, whose backup link resources cannot be shared when their backup mapping paths overlap. Constraint (22) indicates that, under the condition of constraints (20) and (21), if the backup resources are not less than the new backup request, the backup resources can be shared by the new backup request. Otherwise, we must map the backup link and reserve redundant resources.

This ILP is known to be the NP-hard problem, and we propose a new method, SVNLSPM-MST, to solve this ILP of survivable VN link shared protection.

VI. SVNLSPM-MST

A. THE PROCESS OF SVNLSPM-MST

The SVNLSPM-MST which is proposed in this paper, mainly includes an MST link selection algorithm (MSTLS), a virtual node mapping algorithm based on the optimal matching algorithm(OM-VNM), a virtual link mapping algorithm (I-k-SPVLE) and the MST link shared protection algorithm (LSP-MST). The process of SVNLSPM-MST is shown in Fig. 7. First, we calculate the MST topology of VN using the MSTLS and we deal with the node mapping using the OM-VNM. The link mapping starts after the node mapping

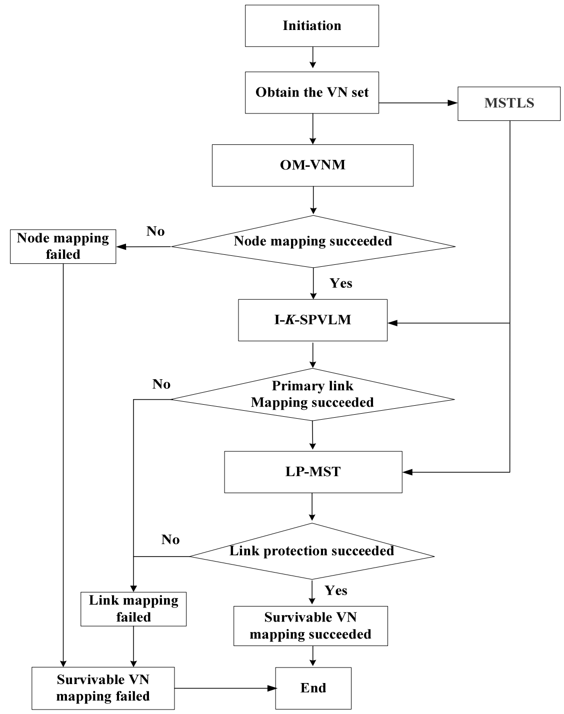


FIGURE 7. The process of SVNLSPM-MST.

is successful. Then we map the virtual primary link using the I-k-SPVLE in the link mapping phase. After that, we use the LSP-MST to map the virtual backup link and share the backup link resources. If the link protection is successful, we judge that the survivable VN mapping is successful, and we reserve resources for the virtual nodes and the virtual links. Otherwise, we judge that the survivable VN mapping fails.

B. MSTLS

In this paper, the improved Kruskal algorithm is adopted to obtain the MST of the VN, and the MSTLS algorithm process is shown in Algorithm 1.

Algorithm 1 MSTLS

Input: G^v
Output: Maximum spanning tree M_{ST}

```
M_ST = {}
for i_1 = 1: n
    select the edge e_i^v ∈ E_P^v whose weight w(e_i^v) is the largest
    if e_i^v can make the M_ST to be an acyclic graph
        save the e_i into the M_ST
    end if
    i_1 = i_1 + 1
end for
```

C. OM-VNM

We propose the OM-VNM to deal with the VN node mapping problem in this paper. The VN node mapping problem is transformed into the optimal matching problem, and then the Kuhn-Munkres algorithm is used to solve it. Compared with the common VN mapping algorithms, OM-VNM algorithm has the following two advantages. First, it prioritizes the matching of the substrate node with the richest node resources and the virtual node with the highest node demand evaluation. This matching limits the mapping area and is beneficial to the load balancing of the SN. Second, we use the ratio of the distances between virtual meta-node and other virtual nodes to the distances between substrate meta-node and other substrate nodes as the matching-degree, the topological connections of nodes are fully considered, which can significantly reduce the mapping link length and the mapping cost.

1) NODE EVALUATION

In this paper, we use the product of the central processing unit (CPU) resources/demand and the adjacent link bandwidth sum to evaluate the nodes. The effect of CPU resources/demand and link bandwidth resources/demand on node evaluation are comprehensively considered, which can be expressed as follows.

$$CB(n) = CPU(n) \cdot \sum_{e \in E} BW(e) \tag{23}$$

Here, $CPU(n)$ represents the CPU resources (demand) of node n, $\sum_{e \in E} BW(e)$ represents the sum of the adjacent link bandwidth. The greater the value of $CB(n)$, the higher the capacity of network node resources/demand will be.

2) OPTIMAL MATCHING PROBLEM

We use the virtual nodes and substrate nodes to construct a weighted binary graph G. As shown in Fig. 8(a), the white nodes and the black nodes represent virtual nodes and substrate nodes, respectively. The VN mapping problem is equivalent to find a matching with the largest sum of edge weights in the weighted binary graph, it is called optimal matching which is solvable in polynomial time. In this paper, the Kuhn-Munkres algorithm [35] is adopted to solve the binary graph optimal matching. First, we transform the weighted binary graph G into a weighted complete binary graph G_c by adding (n-m) zero-nodes which is represented by the gray nodes in Fig. 8(b). Then, we choose the virtual node whose value of node evaluation is the largest as the virtual meta-node and the substrate node whose value of node evaluation is the largest as the substrate meta-node. They are represented by the blue node and the red node, respectively.

The basic idea of the kuhn-munkres algorithm is as follows. First, we take a feasible point label l as the starting point to determine the l -equivalent subgraph G_l. Then, we take any initial matching M_o in G_l, and use the Hungary algorithm to find the maximum matching M_{max}, and judge whether M_{max} is a perfect matching. If so, end of the calculation. Otherwise, the Hungary algorithm is terminated, and the above processes are repeated after the feasible point label is adjusted. If the optimal matching M_b is found, we end of the Kuhn-Munkres algorithm.

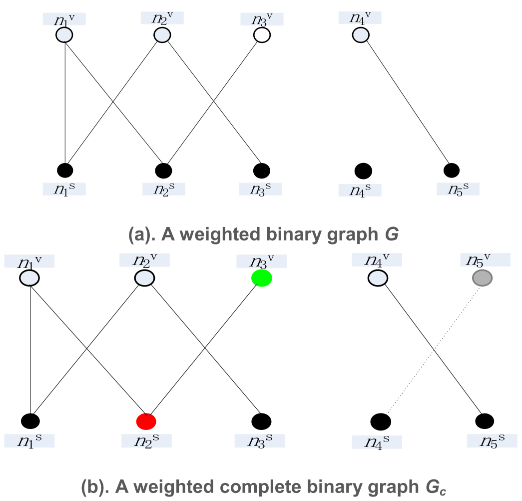


FIGURE 8. The optimal matching problem.

3) NODE MATCHING-DEGREE CALCULATION

Take the distance between virtual meta-node n_I^v and the other virtual nodes n_i^v to the distance between substrate meta-node n_J^s and other substrate nodes n_j^s as the node matching degree P_{ij}, the P_{ij} is defined as the edge weight value of G_c, and it can be defined as follows.

$$P_{ij} = \begin{cases} 1, & \text{if } i = I, \quad j = J; \\ dist(i)/dist(j), & \text{if } j \neq J \& CPU(n_i^v) \leq CPU(n_j^s); \\ 0, & \text{otherwise} \end{cases} \tag{24}$$

In which, I represents the position serial number of the virtual meta-node. J represents the position serial number of the substrate meta-node. $dist(i)$ represents the distance between virtual meta-node n_I^v and other virtual nodes n_i^v, and $dist(j)$ represents the distance between substrate meta-node n_J^s and other substrate nodes n_j^s. $CPU(n_i^v)$ represents the CPU resource demand of virtual node n_i^v, and $CPU(n_j^s)$ represents the CPU resources of substrate node n_j^s. The larger the P_{ij} is, the higher the matching-degree between virtual node and substrate node will be.

According to the equation of matching-degree, the edge weight matrix P of the weighted complete binary graph can be obtained as follows.

$$P = \begin{pmatrix} P_{11}P_{12} & \dots & P_{1n} \\ \dots & \dots & \dots \\ P_{m1}P_{I2} & \dots & P_{mn} \\ \dots & \dots & \dots \\ P_{n1}P_{J2} & \dots & P_{nn} \end{pmatrix} \tag{25}$$

4) SOLUTION OF THE OPTIMAL MATCHING PROBLEM

The Kuhn-Munkres algorithm can be used to directly obtain the optimal matching result M_b of the binary graph G, which is composed of the virtual node and substrate node.

The element in M_b is the set of P_{ij} when the sum of edge weight is maximum. The virtual and substrate nodes connected by the corresponding edge of each P_{ij} are the initial matching of the node mapping. When the CPU resources of substrate nodes do not meet the CPU requirements of virtual nodes, $P_{ij} = 0$. Whereas the optimal matching M_b solved by Kuhn-Munkres algorithm always exists. so zero elements may exist in M_b, and the matching results do not meet the constraints of node mapping. Therefore, a judgment is made before the node mapping. When there are zero elements in M_b, some substrate nodes in the matching result do not meet the CPU requirements of the virtual nodes. Thus, the VN request is rejected and the VN mapping is judged to fail. When there are no zero elements in M_b, the virtual nodes and substrate nodes connected by each P_{ij} corresponding edge are the optimal solution of node mapping, and node mapping is carried out according to this optimal solution. The node mapping process of the OM-VNM algorithm is shown in Algorithm 2.

Algorithm 2 OM-VNM

```
Input: G^s, G^v
output: node mapping list NodemappingList
for n_i^v ∈ N^v
    evaluate n_i^v, and choose n_i^v with the largest resource
    demand as the meta-virtual node n_m^v
end for
for n_j^s ∈ N^s
    evaluate n_j^s and choose n_j^s with the largest resources
    as the meta-substrate node n_m^s
end for
use n_m^s as the optimal matching of n_m^v, and set the
matching degree p to 1
    Use n_i^v and n_j^s to construct a binary graph G
  transform G into a complete binary graph G_c by adding
  (n-m) zero-nodes
    calculate the matching degree P_ij, and generate edge
    weight matrix P of G_c
for i_2 = 1 : n
      take a feasible point label l as the starting point
    to determine the l -equivalent subgraph G_l. Then take
    any initial matching M_o in G_l, and use the Hungary
    algorithm to find the maximum matching M_max
  if M_max is a perfect match
    the match is optimal
  else
    update the feasible point label as l̂
   i_2 = i_2 + 1
  end if
end for
if  there are no zero elements in M_b
  return NODE_MAPPING_SUCCESS
   save the M_b into the NodemappingList
else
    return NODE_MAPPING_FAILED
end if
```

D. I-k-SPVLE

We use the I-k- SPVLE in the link mapping phase, which maps the MST link and NMST links respectively, to avoid overlapping of their mapping paths. The link mapping process is shown in Algorithm 3.

E. LSP-MST

The LSP-MST proposed in this paper divides the virtual links into MST links and NMST links. The MST links are directly protected by the reserved backup resources, and the NMST links are indirectly protected by backup resources by sharing with the MST links. Because the probability of multi-substrate links failing simultaneously is not high, dedicated backup for all virtual links will cause a waste of resources. Thus, the backup links resource shared by different VNs is considered in the LSP-MST. Hence, we design a backup resource sharing mechanism to guarantee that the backup link resources of different VNs can be reasonably shared during the link backup protection process: (i) the mapping path of the primary link and the backup link cannot overlap. (ii) the virtual links of different VNs which are mapped to the same SN path cannot share the same backup resources when their backup link is mapped to the same SN path. (iii) If the backed-up link resources in a substrate link are more than the newly arrived backup link request, the backed-up link resources can be shared. Otherwise, we remap the backup link and occupy the SN resources separately. The process of the LSP-MST is shown in Algorithm 4.

F. COMPLEXITY ANALYSIS

The time complexity of SVNLSPM-MST includes the time complexity of MST link selection, the time complexity of VN node mapping, the time complexity of virtual link mapping and the complexity of backup link mapping. The time complexity of virtual node demand evaluation is $O(|N^v|)$ and the substrate node resource evaluation is $O(|N^s|)$, and the complexity of Kuhn-Munkres algorithm is $O(|N^s|^3)$ under the condition of adding relaxation variables. The time complexity of improved Kruskal algorithm is $O(|E^v| \log |E^v|)$. In the virtual link mapping phase, the improved k-shortest path method is adopted for solution, and the time complexity is $O(\mathrm{k}|N^s|(|E^s| + |N^s|\lg|N^s|))$. In the link backup mapping phase the k-shortest path method is still adopted for solution, and the time complexity is $O(\mathrm{k}|N^s|(|E^s| + |N^s|\lg|N^s|))$. Thus, the total time complexity of SVNLSPM-MST is $O(2\mathrm{k}|N^s|(|E^s| + |N^s|\lg|N^s|) + |N^v| + |N^s| + |N^s|^3 + |E_v|\log |E^v|)$.

VII. SIMULATION

To validate the performance of the SVNLSPM-MST proposed in this paper, we conduct three comparative simulation experiments in this section. The overall performance of the SVNLSPM-MST is compared with another three link

Algorithm 3 I-k- SPVLE

Input: $\boldsymbol{G^s}$,$\boldsymbol{G^v}$, *NodeMappingList*, M_{ST}
Output: link mapping list *LinkMappingList*

rank the bandwidth demand of $e^v_{im} \in M_{ST}$ from high to low to establish the virtual link mapping sequence V_{LMS1}
for $i_3 = 1$: $|M_{ST}|$
  run the k-shortest path algorithm and calculate the candidate substrate path set of the V_{LMS1} one by one
  if $\Omega(e^v_{im}) \in \emptyset$
   return LINK_MAPPING_FAILED
  else
   for $j_1 = 1 : |\Omega(e^v_{im})|$
    map e^v_{im} on the substrate path with the minimum hop count and save the mapping result into *Link_mapping_List1*
    j_1=j_1+1
   end for
  end if
  i_3=i_3+1
end for
rank the bandwidth demand of $e^{\mathbf{v}}_{mp} \in (E^v$-$M_{ST})$ from high to low to establish the virtual link mapping sequence V_{LMS2}
for i_4 =1:$|E^v - M_{ST}|$
  run the k-shortest path algorithm and calculate the candidate substrate path set of the V_{LMS2} one by one
  If $\Omega(e^{\mathbf{v}}_{\mathbf{im}})\in\emptyset$
   return LINK_MAPPING_FAILED
  else
   for $j_2 = 1 : |\Omega(e^{lmd}_i)$-*Link_mapping_List*1$|$
    map the $e^{\mathbf{v}}_{\mathbf{mp}}$ on the substrate link with the minimum hop count and save the mapping result into *Link_mapping_List*2
    J_2=j_2+1
   end for
  end if
  i_4=i_4+1
end for
Link_mapping_List = *Link_mapping_List*1 ∪ *Link_map ping_List*2

protection methods in the Experiment 1. We validate the LSP-MST and OM-VNM, which are the important sub-algorithms of the SVNLSPM-MST in Experiment 2 and Experiment 3, respectively. In the Experiment 2, we validate the performance of LSP-MST on the premise of keeping the original mapping algorithm unchanged. In the Experiment 3, we validate the performance of OM-VNM on the premise of keeping the original link protection algorithm unchanged. The performance evaluation indicators of the link protection methods in each group of experiments are the acceptance ratio of VN mapping, the long-term average revenue to cost ratio and the backup link bandwidth ratio.

Algorithm 4 LSP-MST

Input:M_{ST}, $\boldsymbol{G^s}$, $\boldsymbol{G^v}$, V_{LMS1}, V_{LMS2},B_t(i)
Output: update the VN backup topology mapping result B_t(i+1)

for $i_5 = 1 : |M_{ST}|$
  run the k-shortest path algorithm, map the backup link directly one by one according the V_{LMS1} and backup path e^s_B cannot overlap with the primary path e^s_P
  If the backed-up link resources, $b^{\mathrm{b}} >= b^{\mathrm{new}}$, which is new backup link request
   the backed-up link resources can be shared
  else
   remap the backup link and occupy the SN resources separately
   save the mapping result in $B1$
  end if
  $i_5 = i_5 + 1$
end for
for $i_6 = 1$: $|E(\boldsymbol{G_s}) - M_{\mathrm{ST}}|$
  run the k-shortest path algorithm, map the backup link indirectly one by one according the V_{LMS2} by mapping path sharing without reserving backup resources, and save the mapping results in *B2*
  $i_6 = i_6 + 1$
end for
B_t(i+1) = B_t(i)∪$B1$ ∪ $B2$

A. SIMULATION ENVIRONMENT

In this paper, the SN topology and the VN topology used in the simulation experiment are generated by the improved Salam network topology random generation algorithm [30]. The detailed parameters of SN and VN are shown in Table 2. We use a HP Zhan99 with the Windows 10 operating system to conduct simulation experiments, and the hardware platform is composed of an Intel Core i5-8500 3.0GHz processor with 8GB of RAM. The Matlab R2010a is used to analyze the link protection methods. The duration of simulation experiment is 50,000 time units. A total of 10 simulation experiments are conducted, and the average value is taken as the final simulation result to eliminate the effect of random factors.

B. EXPERIMENT 1: COMPARISON OF DIFFERENT LINK PROTECTION METHODS

In this experiment, the SVNLSPM-MST is compared with the other three virtual link protection methods under the same experimental conditions, as shown in Table 3, and the experimental results are shown in Fig. 9.

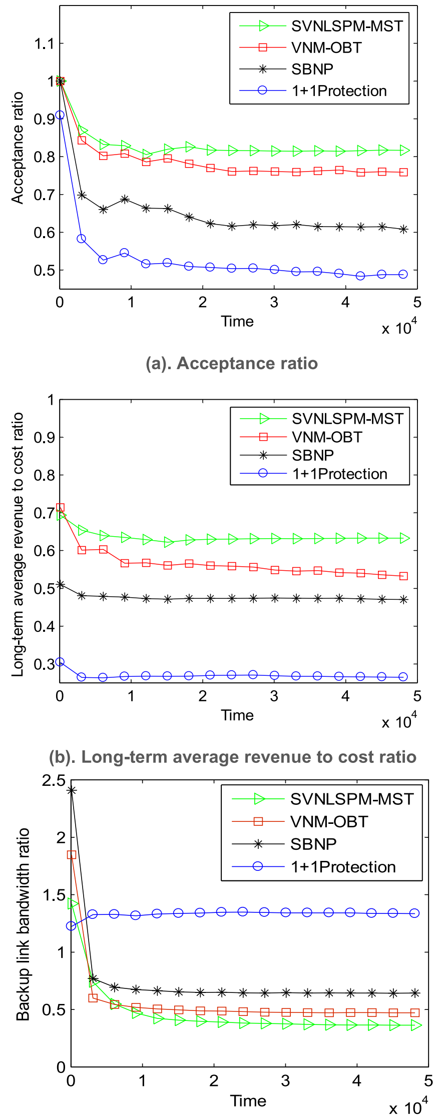
Fig. 9 (a) (b) and (c) respectively illustrate the variation of acceptance ratio, long-term average revenue to cost ratio and backup link bandwidth ratio of the four link protection methods. The acceptance ratio of the 1+1 Protection is maintained at around 0.48, the long-term average revenue to cost ratio is maintained at around 0.26 and the backup

TABLE 2. The detailed parameters of SN and VN.

	SN	VN
node number	100	obey the uniform distribution of [2,15]
link number	500	probability of node connection is 0.5
node CPU	obey the uniform distribution of [50,100]	obey the uniform distribution of [2,20]
link bandwidth	obey the uniform distribution of [50,100]	obey the uniform distribution of [2,20]
VN arrival rate	-	obey the Poisson distribution with parameter 0.05
VN life time	-	obey the exponential distribution with parameter 1500

TABLE 3. Comparision of the link protection methods.

Method	Description
SVNLSPM-MST	The survivable VN link shared protection method based on MST proposed in this paper. The node mapping algorithm is OM-VNM, the link mapping algorithm is I-*k*-SPVLM, and the link protection algorithm is LSP-MST. The backup link resources can be shared.
VNM-OBT	The optimal backup topology link protection method proposed in [25]. The Breadth-First Search algorithm is adopted in the node mapping phase, the *k*-shortest path algorithm is adopted in the link mapping phase. Optimize and simplify the backup topology in the link protection phase and the backup link resources can be shared.
SBNP	The shared on-demand VN link protection algorithm proposed in [24]. The greedy algorithm is adopted in the node mapping phase. The *k*-shortest path algorithm is adopted in the link mapping phase. All virtual links are directly backuped in the link protection phase, and backup link resources can be shared.
1+1Protection	The 1+1 link protection algorithm proposed in [20]. The D-ViNE algorithm is adopted in the node mapping phase, and the *k*-shortest path algorithm is adopted in the link mapping phase. Dedicated backup is adopted to protect all virtual links and link resources cannot be shared.



(a). Acceptance ratio

(b). Long-term average revenue to cost ratio

(c). Backup link bandwidth ratio

FIGURE 9. Comparison of different link protection methods.

link bandwidth ratio is maintained at around 1.35. This is because the D-ViNE algorithm does not fully consider the topological connections between nodes. Thus, the acceptance ratio and long-term average revenue to cost ratio is low and the dedicated backup protection for virtual link causes large resource consumption, the backup link bandwidth ratio is quite high.

The acceptance ratio of the SBNP is maintained at around 0.61, the long-term average revenue to cost ratio is maintained at around 0.46, and the backup link bandwidth ratio is maintained at around 0.7. This is because the greedy algorithm in node mapping does not fully consider the node topology connections, the acceptance ratio and long-term average revenue to cost ratio are not high, and substrate link resource consumption is large. The backup link resources can be shared, so the backup link bandwidth ratio is improved compared with the 1+1 Protection.

The acceptance ratio of the VNM-OBT is maintained at around 0.76, the long-term average revenue to cost ratio is maintained at around 0.53, and the backup link bandwidth ratio is maintained at around 0.49. The VNM-OBT adopts Breadth-First Search algorithm in VN node mapping which fully considers the topological connections of nodes, and thus the acceptance ratio and long-term average revenue to cost ratio is improved. The backup topology is optimized and simplified in the link protection phase, and the backup link resources can be shared. Thus, the backup link bandwidth ratio is significantly reduced.

The acceptance ratio of the SVNLSPM-MST is maintained at around 0.76, the long-term average revenue to cost ratio is maintained at around 0.53, and the backup link bandwidth ratio is maintained at around 0.49. Its performance is best of the four methods. This is because that the OM-VNM adopted in the node mapping phase gives full consideration to the node topological connections, and the LSP-MST adopted in the link protection phase reasonably shares the SN resources. Hence, the substrate resource consumption is reduced and the algorithm has good performance.

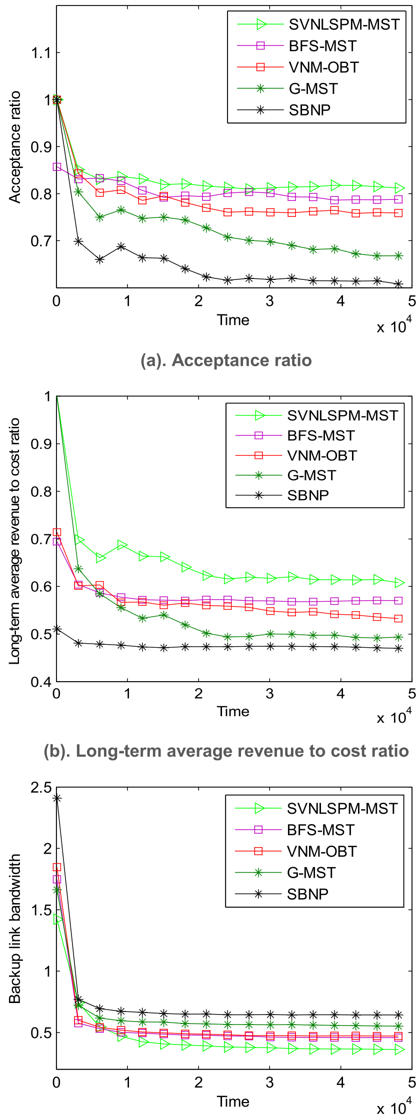
C. EXPERIMENT 2: PERFORMANCE ANALYSIS OF LSP-MST

As shown in Table 4, the LSP-MST is used in this experiment consistently under the condition of keeping the original mapping algorithm of each link protection method unchanged. The result is compared with the Experiment 1 to validate the performance of the LSP-MST. The experimental results are shown in Fig. 10.

TABLE 4. First comparison of the improved methods.

Method	Improved method	Description
SVNLSPM-MST	SVNLSPM-MST	Survivable VN link shared protection method based on MST proposed in this paper. The node mapping algorithm is OM-VNM, the link mapping algorithm is I-*k*-SPVLM, and the link protection algorithm is LSP-MST. The backup link resources can be shared.
VNM-OBT	BFS-MST	The Breadth-First Search algorithm is adopted in the node mapping phase, the *k*-shortest path algorithm is adopted in the link mapping phase, the LSP-MST is adopted for the link protection, and the backup link resources can be shared.
SBNP	G-MST	The greedy algorithm is adopted in the node mapping phase, and the *k*-shortest path algorithm is adopted in the link mapping phase. The LSP-MST is adopted for the link protection, and the backup link resources can be shared.

Fig. 10 (a), (b) and (c) respectively illustrate the variation of acceptance ratio, long-term average revenue to cost ratio and backup link bandwidth ratio of the three improved link protection methods. From Fig. 10, it can be seen that the acceptance ratio of G-MST method remains above 0.67, the long-term average revenue cost ratio remains above 0.50, and the backup link bandwidth ratio remains around 0.65. Compared with the SBNP, the acceptance ratio increases 8%, the long-term average revenue cost ratio increases 9%, and the backup link bandwidth ratio reduces 7%. The acceptance ratio of the BFS-MST method is maintained at around 0.79, the long-term average revenue to cost ratio is maintained at around 0.57, and the backup link bandwidth ratio is maintained at around 0.47. Compared with the VNM-OBT, the acceptance ratio increases 4%, the long-term average revenue to cost ratio increases 7%, and the backup link bandwidth ratio decreases 4%. The acceptance ratio, the long-term average revenue to cost ratio and the backup link bandwidth ratio of the SVNLSPM-MST remain stable maintained at around 0.81, 0.62 and 0.30 respectively, and its performance is the best of the three improved methods.



(a). Acceptance ratio

(b). Long-term average revenue to cost ratio

(c). Backup link bandwidth ratio

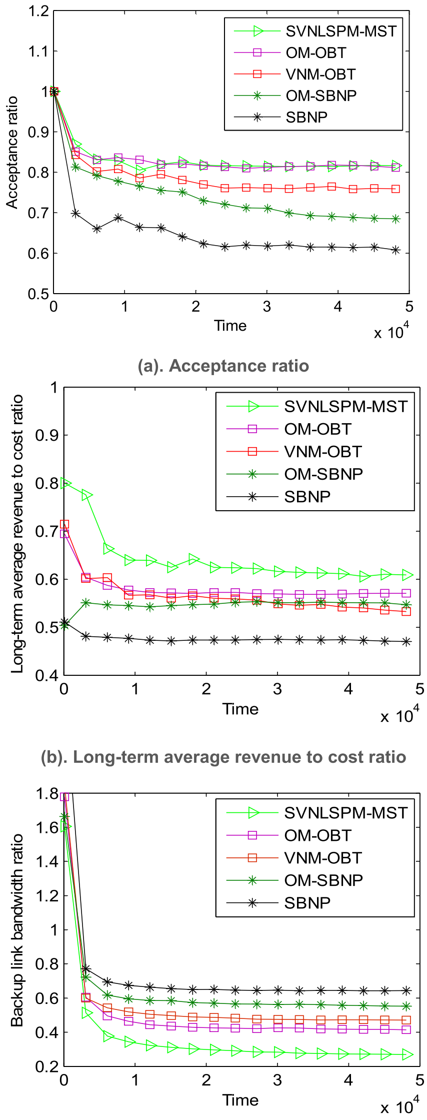
FIGURE 10. First comparison of the improved methods.

This experiment shows that the LSP-MST can reduce the consumption of the substrate resources, and improve performance of link protection.

D. EXPERIMENT 3: PERFORMANCE ANALYSIS OF OM-VNM

As shown in Table 5, the OM-VNM is used in this experiment consistently under the condition of keeping the original link protection algorithm of each link protection method unchanged, and the result is compared with the experiment 1 to validate the performance of the OM-VNM. The experimental results are shown in Fig. 11.

Fig. 11 (a), (b) and (c) respectively illustrate the variation of acceptance ratio, the long-term average revenue to cost ratio and the backup link bandwidth ratio of the three improved link protection methods. It can be seen from the Fig. 11 that the acceptance ratio of the OM-SBNP mapping is maintained at around 0.69, the long-term average revenue to cost ratio is



(a). Acceptance ratio

(b). Long-term average revenue to cost ratio

(c). Backup link bandwidth ratio

FIGURE 11. Second comparison of the improved methods.

maintained at around 0.54, which is 15% and 18% higher than that of the SBNP, respectively, and the backup link bandwidth ratio is maintained at around 0.55, which is 21% lower than that of the SBNP. The acceptance ratio of OM-OBT is maintained at around 0.81, the long-term average revenue to cost ratio is maintained at around 0.57, and these are 6% and 7% higher than that of VNM-OBT, respectively. The backup link bandwidth ratio is maintained at 0.43, which was 12% lower than that of the VNM-OBT. The acceptance ratio, the long-term average revenue to cost ratio and the backup link bandwidth ratio of the SVNLSPM-MST remain stable and keep around 0.81, 0.62 and 0.30, respectively. The performance is best in these three improved methods. This experiment shows that the OM-VNM can reduce the consumption of substrate resources, and improve the mapping performance.

From the above experiments, we can see that the SVNLSPM-MST proposed in this paper performs better than the other link protection methods in terms of the acceptance ratio, the long-term average revenue to cost ratio and the backup link bandwidth ratio. It fully considers the topological connections of nodes in the node mapping phase and reasonably shares the SN resources in the link protection phase. Therefore, the SVNLSPM-MST is a suitable method to deal with the substrate link failures.

TABLE 5. Second comparison of the improved methods.

Method	Improved method	Description
SVNLSPM-MST	SVNLSPM-MST	Survivable VN link shared protection method based on MST proposed in this paper. The node mapping algorithm is OM-VNM, the link mapping algorithm is I-*k*-SPVLM, and the link protection algorithm is LSP-MST. The backup link resources can be shared.
VNM-OBT	OM-OBT	The OM-VNM is adopted in the node mapping phase, the *k*-shortest path algorithm is adopted in the link mapping phase. We optimize and simplify the backup topology in the link protection phase, and the backup link resources can be shared.
SBNP	OM-SBNP	The OM-VNM is adopted in the node mapping phase, the *k*-shortest path algorithm is adopted in the link mapping phase, all virtual links are protected by reserving backup link resources and the backup link resources can be shared.

VIII. CONCLUSION

The survivable VN link protection is studied in this paper, and the SVNLSPM-MST is proposed to handle the substrate link failures. First, a VN mapping algorithm based on optimal matching is proposed to improve the acceptance ratio, and the long-term revenue to cost ratio of VN mapping by considering the topological connections of nodes in the node mapping phase. Second, the MST link protection can ensure the survivability of the VN and reduce the consumption of the substrate resources as much as possible. Then, the reasonable sharing mechanism of the link backup resources can reduce the consumption of the substrate resources. Finally, three experiments are conducted to validate the performance of the SVNLSPM-MST, and the results show that the SVNLSPM-MST has excellent performance compared with the other three link protection methods. The next step is to study the survivable VN node protection against substrate node failures in the future.

REFERENCES

[1] J. Sun, G. Zhu, G. Sun, D. Liao, Y. Li, A. K. Sangaiah, M. Ramachandran, and V. Chang, "A reliability-aware approach for resource efficient virtual network function deployment," *IEEE Access*, vol. 6, no. 1, pp. 18238–18250, Mar. 2018.

[2] T. Anderson, L. Peterson, S. Shenker, and J. Turner, "Overcoming the Internet impasse through virtualization," *Computer*, vol. 38, no. 4, pp. 34–41, Apr. 2005.

[3] H. Cao, H. Hu, Z. Qu, and L. Yang, "Heuristic solutions of virtual network embedding: A survey," *China Commun.*, vol. 15, no. 3, pp. 186–219, Mar. 2018.

[4] A. Fischer, J. F. Botero, M. T. Beck, H. de Meer, and X. Hesselbach, "Virtual network embedding: A survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 4, pp. 1888–1906, 4th Quart., 2013.

[5] I. Pathak and D. P. Vidyarthi, "A model for virtual network embedding across multiple infrastructure providers using genetic algorithm," *Sci. China Inf. Sci.*, vol. 60, no. 4, Apr. 2017, Art. no. 040308.

[6] L. Wang, H. Qu, and J. Zhao, "Virtual network embedding algorithm for load balance with various requests," *Chin. J. Electron.*, vol. 23, no. 2, pp. 382–387, Apr. 2014.

[7] X. Wang, M. Song, D. Yuan, and X. Liu, "Robust virtual network embedding based on component connectivity in large-scale network," *China Commun.*, vol. 14, no. 10, pp. 164–179, Oct. 2017.

[8] H. Cui, W. Gao, and J. Liu, "A virtual network embedding algorithm based on virtual topology connection feature," in *Proc. WPMC*, Atlantic City, NJ, USA, Jun. 2013, pp. 1–5.

[9] D. Jian, H. Tao, W. Jian, L. Jiang, and L. Yunjie, "Virtual network embedding through node connectivity," *J. China Univ. Posts Telecommun.*, vol. 22, no. 1, pp. 17–23, Feb. 2015.

[10] P. Zhang, S. Wu, M. Wang, H. Yao, and Y. Liu, "Topology based reliable virtual network embedding from a QoE perspective," *China Commun.*, vol. 15, no. 10, pp. 46–58, Oct. 2018.

[11] R. R. Oliveira, L. R. Bays, D. S. Marcon, M. C. Neves, and L. S. Buriol, "Dos-resilient virtual networks through multipath embedding and opportunistic recovery," in *Proc. 28th Annu. ACM Symp. Appl. Comput. (SAC)*, Coimbra, Portugal, Mar. 2013, pp. 597–602.

[12] R. R. Oliveira, D. S. Marcon, L. R. Bays, M. C. Neves, L. P. Gaspary, D. Medhi, and M. P. Barcellos, "Opportunistic resilience embedding (ORE): Toward cost-efficient resilient virtual networks," *Comput. Netw.*, vol. 89, no. 7, pp. 59–77, Oct. 2015.

[13] S. Nashid, S. R. Chowdhury, R. Ahmed, A. Khan, S. Fathi, R. Boutaba, J. Mitra, and L. Liu, "Virtual network survivability through joint spare capacity allocation and embedding," *IEEE J. Sel. Area Commun.*, vol. 36, no. 3, pp. 502–518, Mar. 2018.

[14] R. Li, Q. Wu, Y. Tan, and J. Zhang, "On the optimal approach of survivable virtual network embedding in virtualized SDN," *IEICE Trans. Inf. Syst.*, vol. 101, no. 3, pp. 698–708, Mar. 2018.

[15] H. Yu, V. Anand, C. Qiao, and H. Di, "Migration based protection for virtual infrastructure survivability for link failure," in *Proc. OFC/NEOEC*, Los Angeles, CA, USA, Mar. 2011, pp. 1–3.

[16] B. Lu, T. Huang, X. Sun, J.-Y. Chen, and Y.-J. Liu, "Dynamic recovery for survivable virtual network embedding," *J. China Univ. Posts Telecommun.*, vol. 21, no. 3, pp. 77–84, Jun. 2014.

[17] Q. Hu, Y. Wang, and X. Cao, "Survivable network virtualization for single facility node failure: A network flow perspective," *Opt. Switching Netw.*, vol. 10, no. 4, pp. 406–415, Nov. 2013.

[18] A. Jarray and A. Karmouch, "Cost-efficient mapping for fault-tolerant virtual networks," *IEEE Trans. Comput.*, vol. 64, no. 3, pp. 668–681, Mar. 2015.

[19] N. Shahriar, S. R. Chowdhury, R. Ahmed, A. Khan, R. Boutaba, J. Mitra, and L. Liu, "Joint backup capacity allocation and embedding for survivable virtual networks," in *Proc. IFIP*, Stockholm, Sweden, Jun. 2017, pp. 1–9.

[20] S. R. Chowdhury, R. Ahmed, M. M. A. Khan, N. Shahriar, R. Boutaba, J. Mitra, and F. Zeng, "Dedicated protection for survivable virtual network embedding," *IEEE Trans. Netw. Service Manage.*, vol. 13, no. 4, pp. 913–926, Dec. 2013.

[21] M. M. A. Khan, N. Shahriar, R. Ahmed, and R. Boutaba, "Multi-path link embedding for survivability in virtual networks," *IEEE Trans. Netw. Service Manage.*, vol. 13, no. 2, pp. 913–926, Jun. 2016.

[22] T. Guo, N. Wang, K. Moessner, and R. Tafazolli, "Shared backup network provision for virtual network embedding," in *Proc. IEEE ICC*, Kyoto, Japan, Jun. 2011, pp. 1–5.

[23] Z. Wang, J. Wu, Y. Wang, N. Qi, and J. Lan, "Survivable virtual network mapping using optimal backup topology in virtualized SDN," *China Commun.*, vol. 11, no. 2, pp. 26–37, Feb. 2014.

[24] M. R. Rahman and R. Boutaba, "SVNE: Survivable virtual network embedding algorithms for network virtualization," *IEEE Trans. Netw. Service Manag.*, vol. 10, no. 2, pp. 105–118, Jun. 2013.

[25] S. Ayoubi, Y. Chen, and C. Assi, "Towards promoting backup-sharing in survivable virtual network design," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 3218–3231, Oct. 2016.

[26] G. Liu, A. X. An, and S. Su, "Virtual network embedding algorithm with node reliability awareness and shared-path protection," *J. Commun*, vol. 37, no. 8, pp. 51–57, Aug. 2016.

[27] Z. Zhou, T. Lin, and K. Thulasiraman, "Survivable cloud network design against multiple failures through protecting spanning trees," *J. Lightw. Technol.*, vol. 35, no. 2, pp. 288–298, Jan. 15, 2017.

[28] Z. Wang, J. Wu, and D. Cheng, "Coding-aware virtual network embedding for surviving single link failure," in *Proc. IEEE ICC*, Sydney, NSW, Australia, Jun. 2014, pp. 3025–3030.

[29] A. E. Kamal, A. Ramamoorthy, L. Long, and S. Li, "Overlay protection against link failures using network coding," *IEEE/ACM Trans. Netw.*, vol. 19, no. 4, pp. 1071–1084, Aug. 2011.

[30] X. Chen, M. Zhou, S. Zhu, S. Kang, L. Sun, and Z. Zhu, "Optimizing FIPP-*p*-cycle protection design to realize availability-aware elastic optical networks," *IEEE Commun. Lett.*, vol. 22, no. 1, pp. 65–68, Jan. 2018.

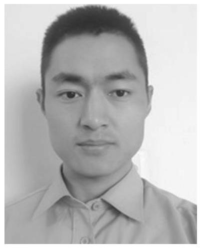
[31] X. Guo, J. Huang, H. Liu, and Y. Chen, "Efficient P-cycle combination protection strategy based on improved genetic algorithm in elastic optical networks," *IET Optoelectron.*, vol. 12, no. 2, pp. 73–79, Apr. 2018.

[32] Y. Su, X. Meng, Q. Kang, and X. Han, "Survivable virtual network link protection method based on network coding and protection circuit," *IEEE Access*, vol. 6, pp. 67477–67493, 2018.

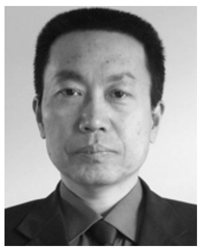
[33] Z. Zhao, X. Meng, Y. Su, Z. Li, "Virtual network embedding based on node connectivity awareness and path integration evaluation," *KSII Trans. Internet Inf. Syst.*, vol. 11, no. 7, pp. 3393–3412, Jul. 2017.

[34] Z. Lu, C. An, and B. Jiao, "An aggregation algorithm for group AHP judgment matrices based on the collective operator of spanning trees," *ACTA Electronica Sinica*, vol. 43, no. 12, pp. 2049–2454, Dec. 2015.

[35] J. Huang and X. Liu, "Design of resource allocation scheme for D2D based on Kuhn-Munkres optimal matching," *Appl. Res. Comput.*, vol. 32, no. 3, pp. 827–829, Mar. 2015.

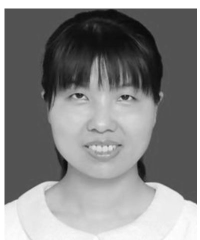


XIAOYANG HAN was born in Pingyu, Henan, China, in 1986. He received the B.S. and M.S. degrees from Air Force Engineering University, China, in 2009 and 2017, respectively, where he is currently pursuing the Ph.D. degree. His research interests include survivable virtual networks and software-defined networks.



XIANGRU MENG was born in Lantian, Shaanxi, China, in 1963. He received the B.S., M.S., and Ph.D. degrees from Xi'an Jiaotong University, China, in 1985, 1988, and 1994, respectively. He is currently a Professor with Air Force Engineering University, Xi'an, China.

From 1995 to 1997, he was a Visiting Scholar with the University of Electronic Science and Technology, Chengdu, China. His research interests include the next-generation Internet, network virtualization, and survivable networks.



QIAOYAN KANG was born in Yongchun, Fujian, China, in 1980. She received the B.S., M.S., and Ph.D. degrees from Air Force Engineering University, Xi'an, China, in 2001, 2004 and 2008, respectively, where she is currently an Assistant Professor. Her research interests include network management and survivable networks.

YUZE SU was born in Jinan, Shandong, China, in 1990. He received the B.S. degree in communication engineering from the Changsha University of Science and Technology, Changsha, in 2013, and the M.S. degree in information and communication engineering from Air Force Engineering University, Xi'an, China, in 2016, where he is currently pursuing the Ph.D. degree in information and communication engineering. His research interests include network virtualization and survivable virtual networks.