Providing survivability for virtual networks against substrate network failure

Ying Wang, Qingyun Chen, Wenjing Li and Xuesong Qiu
State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications
Beijing 100876 -China
[e-mail: wangy@bupt.edu.cn]
*Corresponding author: Ying Wang

Received March 8, 2016; revised June 26, 2016; accepted July 25, 2016; published September 30, 2016

Abstract

Network virtualization has been regarded as a core attribute of the Future Internet. In a network virtualization environment (NVE), multiple heterogeneous virtual networks can coexist on a shared substrate network. Thus, a substrate network failure may affect multiple virtual networks. In this case, it is increasingly critical to provide survivability for the virtual networks against the substrate network failures. Previous research focused on mechanisms that ensure the resilience of the virtual network. However, the resource efficiency is still important to make the mapping scheme practical. In this paper, we study the survivable virtual network embedding mechanisms against substrate link and node failure from the perspective of improving the resource efficiency. For substrate link survivability, we propose a load-balancing and re-configuration strategy to improve the acceptance ratio and bandwidth utilization ratio. For substrate node survivability, we develop a minimum cost heuristic based on a divided network model and a backup resource cost model, which can both satisfy the location constraints of virtual node and increase the sharing degree of the backup resources. Simulations are conducted to evaluate the performance of the solutions. The proposed load balancing and re-configuration strategy for substrate link survivability outperforms other approaches in terms of acceptance ratio and bandwidth utilization ratio. And the proposed minimum cost heuristic for substrate node survivability gets a good performance in term of acceptance ratio.

Keywords: Network virtualization environment; survivable virtual network embedding; fault recovery; resource efficiency

A preliminary version of this paper appeared in IEEE/IFIP NOMS 2014, May 5-9, Krakow, Poland. This version includes a minimum cost survivable virtual network embedding method against substrate node failure. This research was supported by the National High Technology Research and Development Program of China (2013AA013502), and the National Natural Science Foundation of China(61501044).

http://dx.doi.org/10.3837/tiis.2016.09.001 **ISSN : 1976-7277**

1. Introduction

With the evolvement of the Internet, new applications that need more flexible protocols and resource provision, as well as better quality and security assurance are emerging, which brings many challenges to the traditional IP network. Network virtualization is considered as a promising way to support such applications [1]. And a handful of network virtualization prototypes have been developed for specific networking technologies, such as X-Bone for IP networks[2], Tempest targeting ATM networks[3], and Flow-visor[4] for OpenFlow based networks.

In a network virtualization environment, there are three main participants: infrastructure provider (InP), service provider (SP) and end user. A virtual network (VN) is often requested from an end user who does not build and maintain network infrastructure on his own and focuses on the business innovation. Taking a large enterprise with many divisions distributed in different locations as an example, when a virtual network is needed to satisfy some business requirements, the request may involve such information as the computing power of the headquarter and the divisions, the bandwidth capacity between the headquarter and all the divisions, as well as the locations of the headquarter and the divisions. The SP will organize the requirements into a virtual network, which is composed of several virtual nodes and virtual links connecting these nodes. To make it work, the InP will then allocate substrate resources to the virtual network according to the requirements.

When mapping a virtual node or a virtual link, the target substrate node must meet the demands (e.g., CPU, location, storage) of the virtual node. And all of the substrate links along the target substrate path should satisfy the requirements (e.g., bandwidth) of the virtual link. So many constraints make the embedding of virtual networks on a substrate network a challenge in the resource allocation of the virtualization network environment, which is usually referred to as the virtual network embedding (VNE) problem. Much work has been carried out to solve the VNE problem, but few have considered the survivability of VNs (see [5] and references therein).

However, Studies have shown that link and node failures occur as part of everyday operation in networks. Moreover, single failure case happens more often than multiple simultaneous failures. The study [6] states that about 70 % of the unplanned link failures are single link failures. In addition, link failure occurs more frequently than node failure. A study [7] about network-related failures in data centers found out that link failure happens about ten times more than node failure per day. Due to the sharing of substrate resources, a single substrate failure will affect all the VNs sharing it. Furthermore, it will lead to degraded service performance or even service interruption, and will cause the service level agreements (SLA) violations. If no remedy measure is considered beforehand, the InP may not always find available resources for the recovery of the affected VNs and will have to pay the penalties. Hence, survivability mechanisms are required to protect the VNs from such potential frustration.

The purpose of the survivable virtual network embedding (SVNE) is to provide survivability to VNs running on the substrate network against substrate failures through the means of VNE. Since efficient mapping of virtual networks is already a big challenge in the network virtualization environment, the additional survivability requirements undoubtedly increase the difficulty of resource allocation. The main idea of the previous works [8-10] is to allocate backup resources in the virtual network embedding phase to achieve the survivability of VNs. However, the resource efficiency has not been well discussed and evaluated in these works. Because the backup of the VNs takes up a large amount of resources of the InP, it may lead to low acceptance ratio of the VN requests and unacceptable low revenue.

In this paper, we propose two different SVNE mechanisms in the case of single substrate link failure and single substrate node failure respectively. In these mechanisms, different embedding algorithms are proposed to improve the resource efficiency of SVNE. The contributions of our work are:

i. In substrate link failure case, a new linear programming model is formulated to allocate bandwidth resources for the primary paths and the backup paths, by which the embedding scheme of both primary paths and backup paths are solved and the amount of the shared backup bandwidth resources on each substrate link is determined. Also, a more efficient policy of load balancing is proposed to improve the acceptance ratio and the bandwidth utilization ratio.

ii. A more practical survivability mechanism is proposed in case of substrate node failure to deal with the difficulty of achieving the sharing of backup resources while satisfying the location constraints of virtual nodes.

iii. According to the “reserved but not really used” attribute of the backup resources, we propose a reconfiguration heuristic to increase the acceptance ratio and resource utilization ratio of SVNE.

The remaining of this paper is organized as follows. Section 2 reviews related work on VNE and SVNE. In section 3, the problem model is presented. In section 4, a load balancing strategy is proposed and a reconfiguration heuristic is used to solve SVNE problem in substrate link failure scenario. In section 5, a minimum cost strategy to survive substrate node failure is presented. Section 6 shows the results and the analysis of the simulation. Section 7 concludes our work.

2. Related Work

As a core problem of resource allocation in the network virtualization environment, The VNE problem has been studied extensively in previous works (see [5] and its references). The main goal of solving VNE problem is to accept the maximum VNs while using the least substrate resources. Nevertheless, solving the VNE problem is quite difficult because it is NP-Hard even the schedule of VN requests is already known or even the virtual nodes have already been embedded first. To reduce the difficulty of the VNE problem, researchers consider the relaxed version, such as offline [11], and infinite capacity [12]. In [13], to improve acceptance ratio, the authors find the most capable substrate nodes for VN nodes and settle link mapping by supporting path splitting. The authors of [14] use mathematical programming to embed a VN by coordinating node and link mapping.

The main method of providing survivability is to allocate redundant resources for protection before the substrate network failure. In this way, the SVNE can be seen as an embedding for redundant VNs. Hence, the essence of SVNE can be considered the same as that of VNE. Both concern resource allocation, while the SVNE pays more attention to the allocation of backup resources. Therefore, the SVNE problem is much more challenging, since additional backup resources are needed to ensure the intactness of the original VN topology. Much research about SVNE focuses on single substrate link failure or single substrate node failure, which are most popular failure scenarios in everyday network operation.

In the case of substrate link failure, previous works of SVNE problem pay more attention on the way of achieving survivability of virtual networks and the major solution is allocating redundant backup resources when embedding VNs onto the substrate network. In [9], to survive the VNs from single substrate link failure, the bypass paths of the substrate link are preselected before embedding of the VNs to allocate backup bandwidth resources. The backup resources of bypaths can be shared since it assumes one substrate link failure at a time. The authors of [10] propose a profit driven method to overcome single substrate link failure. Economic gain and penalty are modeled for long-term business profit. The goal is to maximize the revenue by balancing the cost and penalty. However, previous works of SVNE against substrate link failure have not fully considered the resource efficiency. While a virtual node only takes up CPU resources on one substrate node, a virtual link takes up bandwidth resources of a substrate path that may consist of several substrate links. The main challenge of providing backup resources in case of substrate link failure is that it may cost too much. In substrate link failure case, the redundant resources may be a big burden to the InP.

The research of substrate node failure in the NVE is much less than that of link failure case. Some representative studies can be seen in [8]. In order to ensure the recovery of important nodes from single substrate node failure, the authors of [8] design a redundant VN topology by adding nodes and links to the original VN. To protect all the important nodes of a VN, 1-redundant scheme of adding one redundant node is proposed, by which the redundant node (backup node) is shared among all the important nodes. In addition, a k-redundant scheme is proposed which provides one exclusive redundant node for each important node in the VN. The authors of [15] study a 1+1-protected virtual network embedding problem, and propose a suite of solutions to guarantee a VN to survive under a single physical node failure. Then, necessary virtual links are added to get the redundant topology of the original VN. The above methods seem to be feasible to recover from single substrate node failure and ensure the intactness of the original VN. However, the location requirement of the backup node has been neglected.

In the case of substrate node failure, the location constraint of backup node is one important factor. Since the embedding of primary virtual nodes considers the location constraint, the neglect of that of backup node is unreasonable. Another difficulty faced in substrate node failure is that when the number of the node which need to be backed up (we call it important node) increases, the amount of redundant virtual links becomes very large. It is important but hard to deal with such an amount of redundant links. To deal with this problem, the sharing degree of the backup bandwidth resources should be raised. However, it is difficult to consider both the sharing of backup nodes and the location constraint. And such scheme has not been proposed yet.

There are also some impressing works about particular failure scenarios in the NVE. The authors of [16][17] study the problem of regional failure resilient virtual network embedding. In [18], the authors propose remapping algorithms of embedding virtual networks to the evolving substrate network because of the adding or deleting of the substrate nodes and links. The work in [19][20] address the problem of survivability of multicast virtual network in datacenter networks. However, the focus of this paper is not these special failure scenarios, but the common scenarios of single substrate link and node failure.

3. Problem Formulation

3.1 Substrate Network and Virtual Network Request

Similar to previous works, we model the substrate network as a weighted undirected graph $G^S=(N^S,E^S)$, where N^s represents the set of substrate nodes, and each substrate node $n_i^s \in N^s$ has attribute values CPU capacity $C(n_i^s)$ and geographic location $LOC(n_i^s)$. E^s

represents the set of substrate links, and each substrate link $e^S(n_u^S, n_v^S) \in E^S$ between substrate nodes n_u^S and n_v^S has an attribute value bandwidth capacity $B(e^S)$. A substrate path $P^S(n_s^S, n_t^S)$ stands for a passable way connecting the source node n_s^S and the terminal node n_t^S. And the bandwidth capacity of the path $B(P^S)$ equals the minimum capacity of all the substrate links constituting P^S.

We also model the virtual network request as a weighted undirected graph $G^V = (N^V, E^V)$. Where, N^V represents the set of virtual nodes. And each virtual node $n_i^V \in N^V$ has attribute values CPU demand $C(n_i^V)$, preferred location $LOC(n_i^V)$ and distance constraint $DIS(n_i^V)$. $DIS(n_i^V)$ stands for the maximum distance allowed from $LOC(n_i^V)$ to the substrate node that virtual node n_i^V is mapped on. E^V represents the set of virtual links and each virtual link $e^V(n_u^V, n_v^V) \in E^V$ has an attribute value bandwidth demand $B(e^V)$.

3.2 VN Embedding

In the network virtualization environment, the VN requests will keep coming. When an VN request arrives, the InP will give a response of accepting it or not. If an VN request is accepted, corresponding substrate resources will be allocated and will not be released until the VN expires. Embedding of a VN request means to find a mapping scheme of the VN request, which satisfies all the required constraints. For node mapping, both the CPU demand and the location constraint should be satisfied and different virtual nodes of a VN request should be mapped on different substrate nodes. For link mapping, the available bandwidth of each substrate link on the target substrate path should be sufficient for the bandwidth demand of the virtual link.

3.3 SVNE Against Single Substrate Link Failure (SVNE-L)

To ensure the successful restoration from single substrate link failure, for any important virtual link, a backup link with bandwidth equals to that of the primary link will also be mapped to a substrate path called backup path. In addition, the substrate links of the primary path and the backup path do not overlap. Different backup paths can share the same backup bandwidth resources on one substrate link. Taking VN request 1 in **Fig. 1** as instance, assuming that link (a, b) is an important link, it has the primary paths embedding $\{(a,b) \to (A - B)\}$, and the backup paths embedding $\{(a,b) \to (A - E - B)\}$ .Without loss of generality, given virtual network request $G^V = (N^V, E^V)$, and substrate network $G^S = (N^S, E^S)$, the SVNE against single substrate link failure (SVNE-L) problem is to map the virtual network to the substrate network to accept maximum VNs with minimum resources while satisfying the follows: (i) for each virtual node/link, it is mapped to the substrate network meeting the capacity/ bandwidth/ location constraints; (ii) the important virtual link is protected against any single link failure of the mapped substrate path.

3.4 SVNE Against Single Substrate Node Failure (SVNE-N)

To ensure the successful restoration from single substrate node failure, when embedding an VN request, for any important virtual node, both the node and the virtual links connected to it should be assigned with prime resources and backup resources. Taking VN request 2 in **Fig. 1** as instance, assume node *c* is an important node, then node *c*, link *(c, d)* and link *(c, e)* should be assigned with prime and backup resources. Node c has the primary node embedding $\{c \to D\}$, and the backup node embedding $\{c \to E\}$. Accordingly link *(c, d)* has the primary paths embedding $\{(c,d) \to (D - E - F)\}$, and the backup paths embedding $\{(c,d) \to (E - F)\}$. And link*(c,e)* has the primary paths embedding $\{(c,e) \to (D - C)\}$, and the backup paths embedding $\{(c,e) \to (E - B - C)\}$. Without loss of generality, given virtual network request $G^V = (N^V, E^V)$, and substrate network $G^S = (N^S, E^S)$, the SVNE against single substrate node failure (SVNE-N) problem is to map the virtual network to the substrate network to accept maximum VNs with minimum resources while satisfying the follows: (i) for each virtual node/link, it is mapped to the substrate network meeting the capacity/bandwidth/location constraint; (ii) the important virtual node is protected against any single substrate node failure. (iii) for each important virtual node (say a) mapped to substrate node m and protected by substrate node n, it satisfies $dis(m, LOC(a)) \le DIS(a)$, and $dis(n, LOC(a)) \le DIS(a)$.

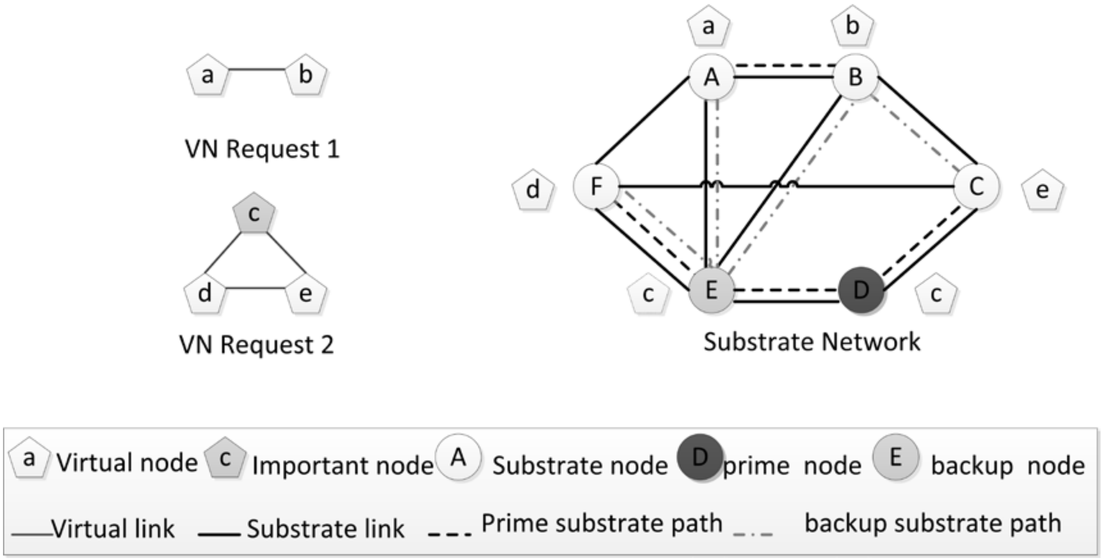

Fig. 1. An instance for SVNE against link and node failures

4. Load balancing and reconfiguration heuristic for SVNE- L problem

The substrate resources will mostly be occupied by continually arriving VN requests. The requirements of the virtual network request are not only about bandwidth of the virtual links but also computing capacity and location requirements of the virtual nodes. With so many constrains, it is difficult to find a mapping solution that meets all these demands. In the case of online VN requests arriving, it is common that there is no feasible solution for a VN request though there are many residual physical resources available. Therefore, both the acceptance ratio and the utilization ratio of substrate network are not very high. In this section, a SVNE scheme based on load balancing and reconfiguration is proposed to survive the VNs from single substrate link failure with high resource efficiency.

4.1 Load Balancing Strategy

Load balancing can be used to improve acceptance ratio by avoiding blocked or bottlenecked area of substrate network that causes rejections of the VN requests. One drawback of load balancing compared with minimum cost strategy is that it may pass more substrate links and cost more. Then, how can we equilibrate the resource cost and the network load balancing? As we know, in SVNE problem, substrate resources could be provisioned as primary resources or backup resources. In addition, usually the backup resources cost much less. For example, the backup paths of virtual links only need reservation of transmission ports. In our method, we allocate backup resources based on load balancing strategy and allocate the primary resources with minimum cost strategy. Thus, the cost of SVNE will not increase much while the substrate network also gets load balanced.

Load balancing has been reflected in [14], but it is the balancing of residual resources of substrate links, which means all the residual resources will tend to be near the average residual value as the VN requests coming continually. Nevertheless, it is unlikely to accept a VN with high bandwidth demand. To deal with this problem, we propose to balance the utilization of substrate links in our mathematical programming model.

In the following part, we present the mathematical programming model of survivable virtual network embedding, which is based on load balancing. In the model, balancing of the utilization ratio of substrate links is designed to increase the acceptance ratio and the utilization ratio of substrate network resources.

Variables:

- $x_l(u,v)$: The resources of substrate link $e^S(u,v)$ allocated for virtual link l as primary bandwidth .
- $y_l(u,v)$: The resources of substrate link $e^S(u,v)$ allocated for virtual link l as backup bandwidth .

Objective:

Minimize:

$$\sum_{u,v\in N^S}\sum_{l\in E^V} x_l(u,v) + \sum_{u,v\in N^S}\sum_{l\in E^V} \frac{B_S(u,v)}{B_r(u,v)+\delta} y_l(u,v) \quad (1)$$

The objective function includes two terms. The first term is to minimize the cost of primary flows. The second term is to balance the utilization of substrate links while mapping backup flows. $B_S(u,v)$ represents the bandwidth capacity of substrate link $e^S(u,v)$. $B_r(u,v)$ represents the residual bandwidth capacity of substrate link $e^S(u,v)$. And δ is a small positive constant to avoid a division by zero. Unlike [14], the coefficient of $y_l(u,v)$ has a correlation with the utilization of the substrate link, and is not just about the residual capacity. In this way, the utilization ratio of substrate links will increase gradually and low residual capacity of all substrate links can be avoided. With these two terms in the objective function, the primary resources will be allocated with minimum cost strategy, and the backup resources will be allocated based on load balancing strategy.

Constraints:

Constraints of primary and backup flows:

$$\sum_{n\in N^S} x_l(s_l,n) - \sum_{n\in N^S} x_l(n,s_l) = b_l, \forall l \in E^V \quad (2)$$

$$\sum_{n\in N^S} y_l(s_l,n) - \sum_{n\in N^S} y_l(n,s_l) = b_l, \forall l \in E^V \quad (3)$$

$$\sum_{n\in N^S} x_l(n,t_l) - \sum_{n\in N^S} x_l(t_l,n) = b_l, \forall l \in E^V \quad (4)$$

$$\sum_{n \in N^S} y_l(n, t_l) - \sum_{n \in N^S} y_l(t_l, n) = b_l, \forall l \in E^V \quad (5)$$

$$\begin{aligned} &\sum_{n \in N^S} x_l(n, i) - \sum_{n \in N^S} x_l(i, n) = 0, \\ &\forall l \in E^V, \forall i \in N^S \setminus \{s_l, t_l\} \end{aligned} \quad (6)$$

$$\begin{aligned} &\sum_{n \in N^S} y_l(n, i) - \sum_{n \in N^S} y_l(i, n) = 0, \\ &\forall l \in E^V, \forall i \in N^S \setminus \{s_l, t_l\} \end{aligned} \quad (7)$$

Formula (2) to (7) are constraints for primary flows and backup flows.

Formula (2) is the constraint of primary flows for the source nodes. It denotes that for any virtual link $l \in E^V$, at the source node of the primary substrate path that l is embedded on, the difference between the input flow and the output flow is exactly the bandwidth requirement of virtual link l. Accordingly, Formula (3) is the constraint of backup flows for the source nodes. b_l represents the bandwidth requirement of virtual link l, and s_l represents the source node of the primary or backup substrate path that virtual link l is embedded on.

Formula (4) is the constraint of primary flows for the terminal nodes. It expresses that for any virtual link $l \in E^V$, at the terminal node of the primary substrate path that l is embedded on, the difference between the output flow and the input flow is exactly the bandwidth requirement of virtual link l. Similarly, Formula (5) is the constraint of backup flows for the terminal nodes. t_l represents the terminal node of the primary or backup substrate path that virtual link l is embedded on.

Formula (6) is the constraint of the middle nodes that primary flows pass through. It denotes that for any virtual link $l \in E^V$, at any substrate node that is neither the source node nor the terminal node of the primary substrate path that l is embedded on, the input flow and the output flow is equal. Accordingly, Formula (7) is the constraint of the middle nodes that backup flows pass through. i represents the middle node of the primary or backup substrate path that virtual link l is embedded on .

Constraint of no overlaps:

$$\begin{aligned} &x_l(m, n) + y_l(m, n) \le b_l, \\ &\forall l \in E^V, \forall m \in N^S, \forall n \in N^S \end{aligned} \quad (8)$$

Formula (8) denotes that for any virtual link $l \in E^V$, the sum of the primary bandwidth and backup bandwidth allocated for l from a substrate link $e^S(m, n)$ is less than or equal to the bandwidth requirement of virtual link l. It guarantees that the primary flow and backup flow of a virtual link are on two different substrate paths.

Constraint of capacity and backup share:

$$\begin{aligned} &\sum_{l \in E^V} x_l(m, n) + \max_{l \in E^V}\{y_l(m, n), Y(m, n)\} \\ &\le B_r(m, n) + Y(m, n), \forall m \in N^S, \forall n \in N^S \end{aligned} \quad (9)$$

Formula (9) expresses that the sum of the primary and backup bandwidth resource that are allocated for l from the substrate link $e^S(m, n)$ is less than or equal to the sum of the residual resources and the backup resources that has been allocated on the substrate link $e^S(m, n)$. In

Formula (9), $Y(m,n)$ represents the amount of backup resources that has been allocated on the substrate link between m and n. The function max is to select the max backup flow on a substrate link. If it is bigger than $Y(m,n)$, the value of $Y(m,n)$ will be replaced by the new max backup flow. $B_r(m,n)$ is the residual bandwidth capacity of substrate link $e^S(m,n)$. In this way, Formula (9) constrains the bandwidth capacity and decides the amount of shared backup resources altogether.

Domain constraints:

$$x_l(m,n) \geq 0, \forall l \in E^V, \forall m \in N^S, \forall n \in N^S \qquad (10)$$

$$y_l(m,n) \geq 0, \forall l \in E^V, \forall m \in N^S, \forall n \in N^S \qquad (11)$$

Formula (10) and (11) are non-negative constraints for variables $x_l(m,n)$ and $y_l(m,n)$, which denote that either the primary bandwidth or backup bandwidth allocated for any virtual link l from any substrate link $e^S(m,n)$ is non-negative.

4.2 Reconfiguration Heuristic

Different from the primary resources, the backup resources are only reserved and not activated before any failure occurrence, which makes reconfiguration of backup resources easier, cheaper and less risky. Therefore, we can reconfigure the backup resources dynamically as needed to get more resources used for embedding, and get the VN requests more likely to be accepted. So, in some cases, we can try to change the mapping of the backup flows to accept some VNs which cannot be accepted at first. This idea, to our best knowledge, is used in the SVNE problem for the first time. Based on the above analysis and considerations, we propose a heuristic algorithm to survive the substrate link failure and improve the resource efficiency through load balancing and reconfiguration.

The detail of the algorithm is shown in **Fig. 2**. At first, the virtual nodes of the VN request are embedded using the algorithm 1 proposed in [13]. Then, both the primary bandwidth and the backup bandwidth resources are allocated to the virtual links of the VN request through solving the linear programming model proposed in Section 4.1. If no solution could be found, the algorithm will take the previous assigned backup resources as available resources and solve the linear programing model again. If a solution could be found, the backup resources will then be reconfigured and the mapping solution will be output. The essence of our algorithm is trying to improve the resource efficiency by load balancing the utilization of the substrate resources and reconfiguring the backup resources. Load balancing can improve acceptance ratio by avoiding blocked area and tiny resource fragments of substrate network. Since the reconfiguration of backup resources is easier, cheaper and less risky, it will be more efficient to dynamically control the backup resources. Our proposed algorithm is an application of this idea, which reconfigures the backup links to accept the VNs rejected at first.

Algorithm 1: Heuristic algorithm to survive substrate link failure

```
Input: Virtual Network Request $G^V = (N^V, E^V)$
Output: Mapping solutions for virtual nodes, primary links and backup links
    Get state of Substrate Network $G^S = (N^S, E^S)$
    for $n_i^V \in N^V$
```

```
        embed $n_i^V$ with Algorithm 1 in [13]
        if demands cannot be satisfied
            return false
        end if
    end for // node embedding
    Solve the linear programming model proposed in section 4.1
    if no solution found
        for $e^S(m,n) \in E^S$ do
            $Y(m,n)=0$
        end for // take the backup resources as available resources
        solve the linear programming model again
        if no solution found
            return false
        else reconfigure the backup resources
            return true
```

Fig. 2. Heuristic algorithm to survive substrate link failure based on load balancing and reconfiguration

5. Minimum cost heuristic for SVNE-N problem

The scenario of single substrate node failure is much more intractable than that of single substrate link failure. In the latter case, only the important virtual links need to be protected. However, in the former case, both the important virtual nodes and the virtual links connected to them should be protected. Therefore, when dealing with the backup of important nodes of the virtual network, both the backup node resources and the bandwidth resources should be pre-allocated for failure recovery.

When choosing substrate nodes for the backup of virtual nodes, previous works have neglected the location constraints of these virtual nodes to achieve the sharing of backup resources [8][10]. Nevertheless, it makes no sense that considering the location constraints when allocating primary resources but ignoring these when allocating backup resources. This way the location constraint of the important virtual node will not be satisfied after migration from the primary substrate node to the backup substrate node. For cross-region business, different virtual nodes in a virtual network usually have different location preference. To solve this problem, we split the original substrate network into two parts, the network of remaining resources and the network of backup resources. An example of the network segmentations is shown in **Fig. 3**.

In **Fig. 3**, the network on the left is the available remaining resources of the substrate network, which can be used as primary resources and backup resources for the coming VN requests. The network on the right is the backup resources that have been allocated to the VNs that running on the substrate network. And the backup resources are shared by different important virtual nodes and redundant virtual links of different virtual networks. For example, on substrate node A, there are 25 units of CPU resources left and no backup resources. On substrate node D, there are 25 units of available CPU resources left and 10 units of backup CPU resources that shared by VNs running on the substrate network. In addition, there are 30 units of bandwidth resources left and 20 units of backup bandwidth resources on the substrate link B-E. When embedding a VN request, primary resources are allocated by the network of remaining resources. Then the backup resources for important nodes and the additional links will be allocated. From such segmentation, the location constraint for both prime nodes and backup nodes could easily be satisfied.

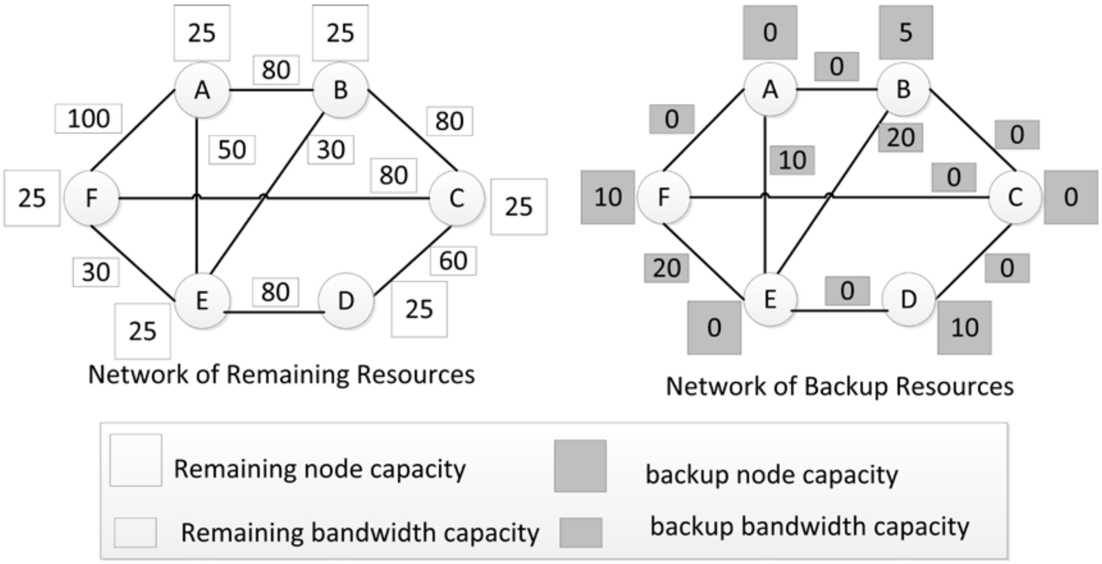
25
25
80
A
B
100
50
30
80
80
25
F
C
25
30
60
E
80
D
25
25
Network of Remaining Resources
0
5
0
A
B
0
10
20
0
10
F
0
C
0
20
0
E
0
D
0
10
Network of Backup Resources
Remaining node capacity
backup node capacity
Remaining bandwidth capacity
backup bandwidth capacity

Fig. 3. Divided networks of the substrate network

In order to improve the degree of resources sharing, we transfer this objective to minimize the cost of backup resources. And we define a cost model of backup resources as follow:

$$f_n^{\,m} = \begin{cases} C_m - B_n, C_m \geq B_n \\ 0, C_m < B_n \end{cases} \tag{12}$$

In Eq. 12, $f_n^{\,m}$ is the cost of backup resources for a virtual node or a link m when it is embedded on a substrate node or link n. C_m is the resource demands of the virtual node or link. B_n is the backup resources already allocated on the substrate node or link. When the amount of backup resources of the substrate node or link is bigger than the need of backup resources of the virtual node or link, the cost will be 0. For example, in **Fig. 3**, if there is a virtual node with a CPU requirement of no more than 10 and its backup node is substrate node D, there will be no backup cost of this node. Else, the backup cost will be the difference between the backup resources needed and the backup resources that already allocated. For example, if there is a virtual link with a bandwidth requirement of 15 and substrate link *A-E* is on its backup path, the backup cost of substrate link *A-E* for the virtual link will be 5.

Based on the above cost model, we can get an objective function of the minimum cost of backup resources, and the cost can be easily worked out. In substrate node failure case, the backup process can be in a two-stage way. First, for all the important virtual nodes in the VN request, find the backup nodes satisfying their location constraints so that the total cost can be minimized by the cost model.

$$min \sum_{m \in N^B, n \in \Omega(m)} f_n^{\,m} \tag{13}$$

$$dis(n, LOC(m)) \leq DIS(m), \forall n \in \Omega(m) \tag{14}$$

$$min \sum_{v \in E^B, l \in E^S} f_l^v \tag{15}$$

In Formula 13, N^B are the set of important nodes of the virtual network that need to be backed up and $\Omega(m)$ are the substrate nodes that satisfy the preferred location constraints of virtual node m. $\Omega(m)$ could be figured out through the following formula (14). Then minimize the cost of redundant virtual links by the following objective. In Formula 15, E^S are the substrate links and E^B are the redundant links, and f_l^v is the cost of backup resources for virtual link v when it is embedded on substrate link l.

Fig. 4 shows the detail of our proposed algorithm. Firstly, the original VN is embedded with the classical method (Algorithm 1 for node embedding and Algorithm 2 for link embedding) in [13]. Then the backup nodes and links are embedded with the minimum cost defined by the cost model, during which the location constraints of the backup nodes are satisfied and the sharing degree of backup resources are optimized.

Algorithm 2: Heuristic algorithm to survive substrate node failure

```
Input: Virtual Network Request $G^V = (N^V, E^V)$ , backup nodes $N^B \subseteq N^V$
Output: Mapping solutions for virtual nodes, backup nodes, primary links and redundant links
Get state of Substrate Network $G^S = (N^S, E^S)$
embed $G^V$ with Algorithm 1 and Algorithm 2 in [13]
if embedding fails
    return false // primary resource allocation
for $n_i \in N^B$
    find a substrate backup node with minimum cost
    if demands cannot be satisfied
        return false
    end if
end for // backup node embedding
get the redundant links $E^B$ for node restoration
for $e_i \in E^B$
    find a substrate backup path with minimum cost
    if demands cannot be satisfied
        return false
    end if
end for // redundant link embedding
```

Fig. 4. Heuristic algorithm to survive substrate node failure based on minimum cost

6. Performance Evaluation and Analysis

In this section， the simulation environment is described and the performance of the proposed algorithms is also analyzed.

6.1 Simulation Environment

We develop a simulator of network virtualization environment for survivable virtual network embedding. We setup the simulation and choose the parameters in accordance with the previous works on SVNE or VNE problem [8] [10] [13]. We generate the topology of substrate network and virtual network by the GT-ITM tool [21]. We generate the substrate network with 100 nodes and 500 links, equivalent to the scale of a medium-sized ISP network. The CPU capacity of the substrate nodes follows a uniform distribution between 50 and 100 resource units. And the location of these nodes $LOC(n^V)$ are equally distributed in a [100,100] area. The bandwidth capacity of the substrate links are real numbers uniformly distributed between 0 and 100. For virtual network simulation, the arrival of VN requests follows Poisson process with an average rate of one VN per 3 time units, and the duration time of the VN requests follows an exponential distribution with an average of 100 time units. The number of virtual nodes in a VN request is ranging in [2, 9] and the average connectivity ratio between two virtual nodes is 50%. We randomly select the preferred location of each node in a [100,100] area, and set the distance constraint $DIS(n^V)$ to 20. We uniformly select the CPU demands of virtual nodes between (0, 30] and uniformly select the bandwidth demands of virtual links between (0, 50]. The linear program model is solved by the GLPK tool [22].

6.2 Metrics and Comparison Method

a）. Average acceptance ratio

The average acceptance ratio is the proportion of successfully embedded virtual networks in the arrived virtual networks at time t.

b）. Average cost

The cost of a VN request is based on the substrate resources allocated to it and its duration time. In SVNE embedding case, there will be extra cost of backup resources. Moreover, usually the operation cost of backup resources is less than that of primary resources. The total cost of embedding a VN request with survivability can be formulated as follow:

$$Cost(G^V) = T(G^V)(\sum_{l\in E^V}\sum_{s\in E^S} F_w(l,s) + \beta\sum_{l\in E^V}\sum_{s\in E^S} F_b(l,s) + \alpha\sum_{n\in N^V} C(n)) \qquad (16)$$

Where $T(G^V)$ is the duration time of the VN request; $F_w(l,s)$ is the primary bandwidth resources allocated for virtual link l on substrate link s; $F_b(l,s)$ is the backup bandwidth resources allocated for virtual link l on substrate link s; $C(n)$ is the CPU resource allocated for virtual node n; $0 \le \beta \le 1$ is a coefficient that weights the cost of backup resources; and α is the relative weight between CPU and bandwidth resources.

c）. Average utilization ratio of substrate resource

The average utilization ratio of substrate resource is the amount of assigned substrate resource capacity divided by the sum of available substrate resource capacity at time t.

d）. Average backup resource occupation ratio

The average backup resource occupation ratio is the amount of backup resource capacity divided by the sum of available substrate resource capacity at time t.
In the case of substrate link failure, we compare four strategies. The compared approaches for SVNE against link failure are illustrated in **Table 1**.

Table 1. Compared SVNE approaches against link failure

Scheme	Strategy used in link embedding
SP	Shortest path.
LB_R	Load balancing of residual bandwidth capacity of the substrate links [14].
LB_U	Load balancing of utilization of the substrate links.
LB_U_RM	Load balancing of utilization of the substrate links and reconfiguration of backup links.

In the case of substrate node failure, we compare the results of different numbers of backup nodes with our proposed algorithm. In addition, we compare the performance of different strategies (minimum cost and load balancing) with the same number of backup nodes. The compared approaches for SVNE against node failure are illustrated in **Table 2**.

Table 2. Compared SVNE approaches against node failure

Scheme	Strategy used in backup resources allocation and the setting of important node
NB_M	None of virtual nodes need to be backed up The VN requests are embedded with minimum cost strategy.
NB_L	None of virtual nodes need to be backed up The VN requests are embedded with load balancing strategy.
HB_M	Half of nodes of the VN request are randomly selected as important nodes The backup resources of important nodes and redundant links are allocated with minimum cost strategy.
HB_L	Half of nodes of the VN request are randomly selected as important nodes The backup resources of important nodes and redundant links are allocated with load balancing strategy.
FB_M	All nodes of the VN request are important nodes The backup resources of important nodes and redundant links are allocated with minimum cost strategy
FB_L	All nodes of the VN request are important nodes The backup resources of important nodes and redundant links are allocated with load balancing strategy

6.3 Simulation Results and Analysis

a）. The efficiency of SVNE- L method

In the case of substrate link failure, we compare the performance of the four strategies. We compute the average acceptance ratio (**Fig. 5**) and the average utilization of substrate link (**Fig. 6**). We also study the average cost (**Fig. 7**) and the average backup bandwidth occupation (**Fig. 8**). From Fig.5 to Fig.8, we have the following observations.

1) Load balancing strategy gains higher acceptance ratio than shortest path strategy. And load balancing of utilization is better than load balancing of residual resources. Among the four schemes, the acceptance ratio of LB-U-RM is the highest.

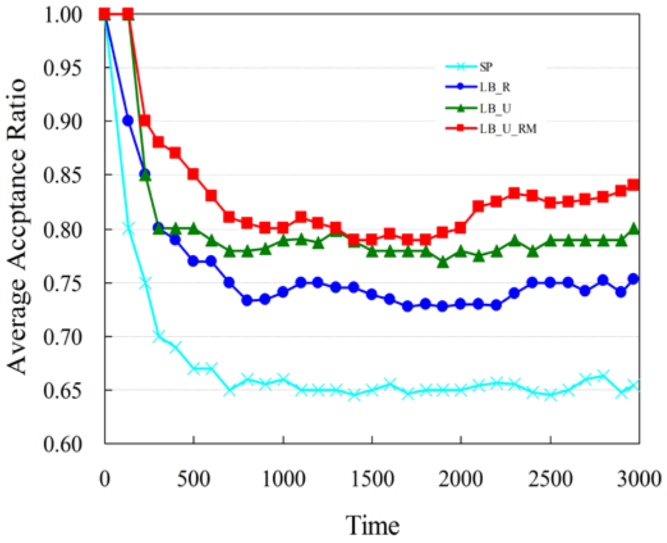

Fig. 5. Average acceptance ratio over time

Fig. 5 presents the acceptance ratio of the four schemes for substrate link failure case during the simulation. As time goes by, the acceptance ratio gets steady. As we can see, the acceptance ratio of load balancing strategy is higher than shortest path (SP) strategy. Compared with the load balancing case, shortest path strategy may make some substrate paths more blocked and therefore harder to accept more VNs.

It can also be observed that load balancing of utilization (LB_U and LB_U_RM) is better than load balancing of residual resources (LB_R). Compared with utilization balancing, it is harder for residual bandwidth capacity balancing to find paths for virtual links with bigger bandwidth demands, because the residual bandwidth of all substrate links decrease synchronously when the bandwidth is saturated. Among the four methods, the acceptance ratio of LB_U_RM is the highest. It indicates that the reconfiguration of backup links does help to increase the chances of accepting more VNs.

2) Load balancing gets higher utilization ratio of substrate links than that of shortest path. Though load balancing increases the utilization ratio, the average value is not very high. Among the four schemes, the utilization ratio of LB-U-RM is the highest.

Fig. 6 shows the utilization ratio of the substrate links. As we can see, it is quite hard to increase the utilization ratio of substrate links because of the complexity of the VN demands, which are not only about resources but also about location preference and topology. One result is that there exist many pieces of available bandwidth resources. When dealing with backup resources, it is helpful to increase the utilization ratio by making use of these available resources. High utilization of substrate resources can save the resource cost for InPs. So if backup resources can be handled in proper way, providing backup resources cannot be seen as a total burden for an InP. Since the utilization of substrate network is not high, there is a big chance to accept the VN request rejected at first by remapping backup paths. The acceptance ratio (**Fig. 5**) and the utilization ratio (**Fig. 6**) of LB_U_RM indicate the effectiveness of our proposed algorithm.

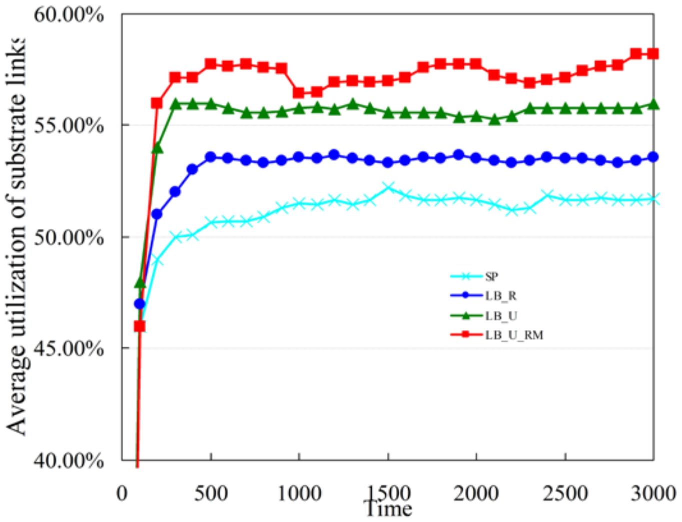

Fig. 6. Average utilization of substrate links

3) Schemes with higher acceptance ratio cost a bit more and gets higher backup bandwidth occupation ratio. But the differences are not big.

The average cost is shown in **Fig. 7**, and the average percentage of the backup resources occupation is presented **Fig. 8**. Strategy with higher acceptance ratio will need more backup resources and cost a bit more because more backup resources are allocated. However, as could be observed from **Fig. 8**, the differences of backup bandwidth occupation ratio among 4 schemes are not big.

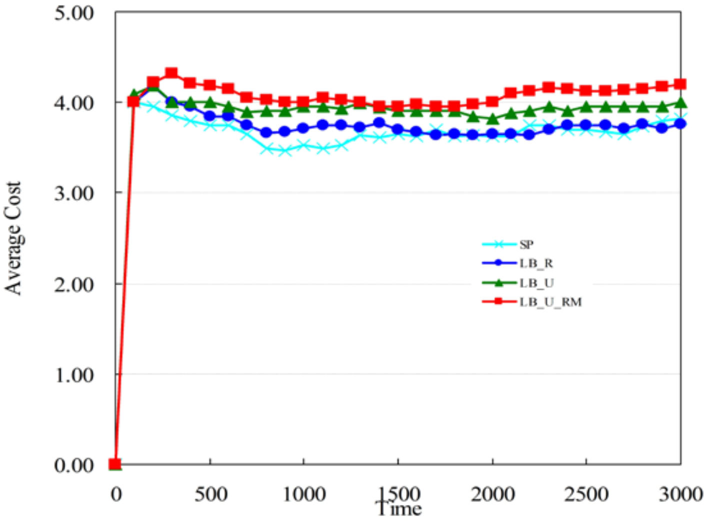

Fig. 7. Average cost

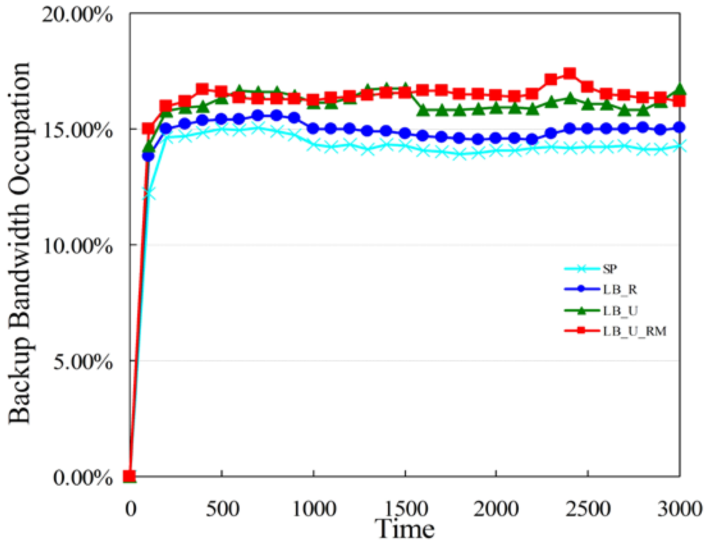

Fig. 8. Average backup bandwidth occupation

b）.The efficiency of SVNE- N method

In the case of substrate node failure, we firstly compare minimum cost and load balancing strategy given the same number of important nodes, and compute the acceptance ratio (**Fig. 9**). Then we study our minimum cost strategy with different number of important nodes and compute the acceptance ratio (**Fig. 10**). We also evaluate the backup resource occupation ratio of our minimum cost scheme (**Fig. 11**). From **Fig. 9** to **Fig. 11**, we have the following observations.

1) Minimum cost scheme gets higher acceptance ratio when the number of important nodes increases.

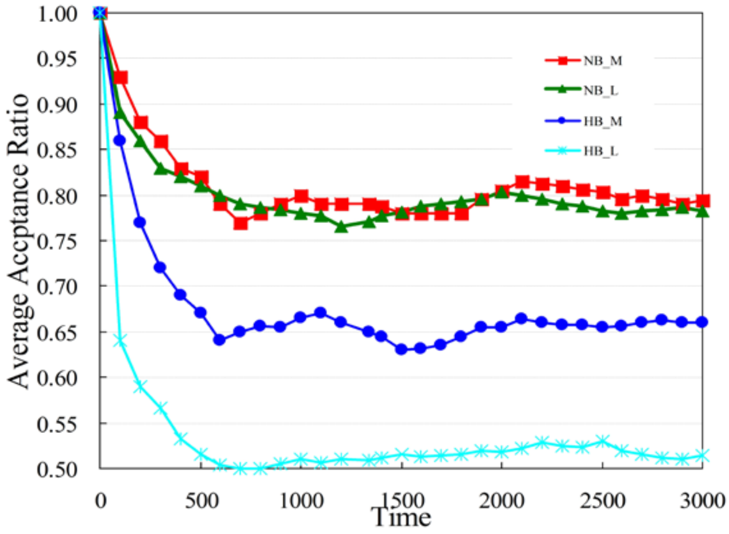

Fig. 9. Average acceptance ratio of load balancing and minimum cost

In **Fig. 9**, we compare the acceptance ratio of different number of important nodes with minimum cost and load balancing strategy. In no backup case, which is VN embedding, load balancing (NB_L) and minimum cost (NB_M) have almost the same acceptance ratio. However, when more nodes need to be backed up, the situation changes. As illustrated in **Fig. 9**, when half of virtual nodes need to be backed up, the acceptance ratio of load balancing gets much lower. As the important nodes increase, the requirements of backup resources increase too. In this case, the minimum cost scheme could get higher sharing degree of backup resources, and save more resources to accept new VN request, thus could improve the acceptance ratio.

2) The acceptance ratio decreases as the number of important nodes increases.

Simulation results of different number of important nodes are shown in **Fig. 10**.

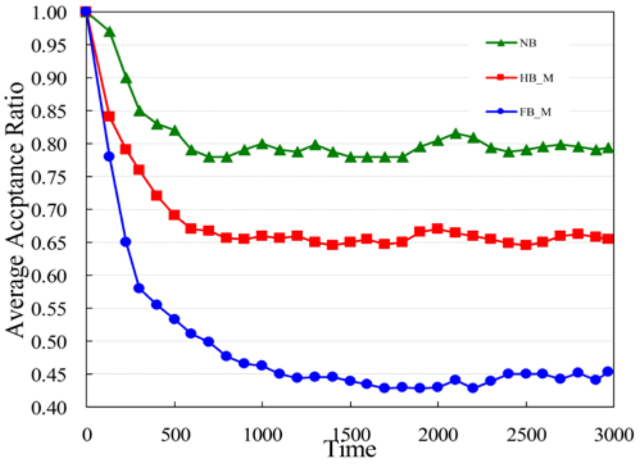

Fig. 10. Average acceptance ratio of different number of backup nodes

As we can see, when the number of important nodes increases, the decrease of acceptance ratio is obvious. It is because that as the number of important nodes increases, the need of backup computing resources and redundant bandwidth resources will increase significantly. And this is the most important problem faced in dealing with substrate node failure.

3) Backup resource occupation ratio of backup link is higher than that of backup nodes

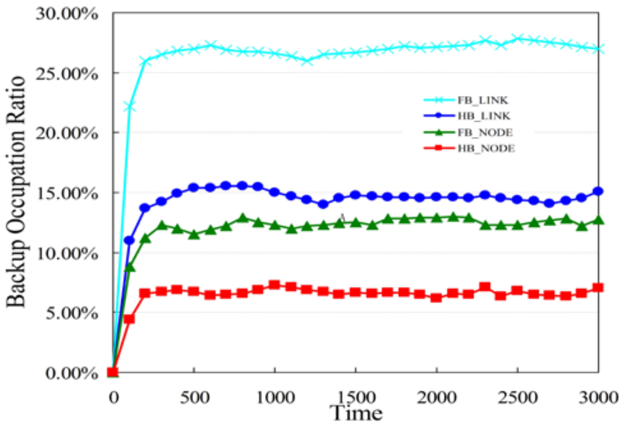

Fig. 11. Backup resource occupation ratio: Backup resource occupation ratio of backup link is higher than that of backup nodes.

Fig. 11 shows the proportion of backup resource in substrate node failure case. The occupation ratio of backup link resource is much higher than that of backup node resources. This is mainly because embedding of a virtual link need more than one substrate link. Link embedding is always the important part of virtual network embedding, and it concerns to the performance of embedding.

7. Conclusion

Survivability is an important capability for a network to operate normally. In the network virtualization environment, owing to the loose coupling between virtual resources and substrate resources, it is possible to gain the survivability of virtual networks against substrate failures from VNE point of view, which is known as survivable VNE. Since single substrate link failure and single substrate node failure are the common failure scenarios in everyday network operation. In this paper, we dwell on the SVNE against such failure scenarios separately from the perspective of improving the resource efficiency.

In the case of single substrate link failure, we propose a linear programming model, which employs a more efficient strategy of load balancing. Such load balancing strategy can improve acceptance ratio by avoiding blocked area and tiny resource fragments of substrate network. Besides, According to the “reserved but not really used” attribute of the backup resources, we propose a reconfiguration strategy, which can dynamically take the backup resources as available resources and increase the chance of accepting the VN request that have been rejected first. This idea is applied in our heuristic algorithm to solve the SVNE against substrate link failure in this paper. And it could also be used to other SVNE problems.

In the case of single substrate node failure, to achieve higher sharing degree of backup resources while satisfying the location constraints of backup nodes, we firstly split the substrate network into a remaining network and a backup network. So that the primary and backup resources are allocated separately while both the location constraint and the sharing of the backup resources can be satisfied. Then, we propose a cost model of backup resources, which is used by a heuristic to minimize the cost of backup resources.

Simulations are conducted and the performance of the proposed methods is validated by the results. The proposed load balancing and re-configuration strategy for substrate link survivability outperform other approaches in terms of acceptance ratio and bandwidth efficiency. And the proposed minimum cost heuristic for substrate node survivability gets a good performance in term of acceptance ratio. In the future, we plan to extend the proposed methods to solve SVNE problem in more complex failure scenarios, such as multiple substrate failures, and regional failure.

References

[1] Chowdhury N K and Boutaba R, “Network virtualization: State of the art and research challenges,” *IEEE Communications Magazine*, vol. 47, no. 7, pp. 20–26, July 2009. Article (CrossRef Link).

[2] Touch J, “Dynamic Internet Overlay Deployment and Management using the X-Bone,” *Computer Networks*, vol.36, no. 2–3, pp. 117–35, July 2001. Article (CrossRef Link).

[3] J. E. van der Merwe et al., “The Tempest: A Practical Framework for Network Programmability,” *IEEE Network*, vol. 12, no. 3, pp. 20–28, May/June 1998. Article (CrossRef Link).

[4] Salvadori Elio, et al., “Demonstrating generalized virtual topologies in an openflow network,”*ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4, pp. 458 – 459, Oct. 2011. Article (CrossRef Link).

[5] Fischer A, Botero J F , Beck M T , DeMeer H, and Hesselbach X , “Virtual Network Embedding: A Survey,” *IEEE Communications Surveys & Tutorials*, vol.15, no.4, pp. 1888 - 1906, FOURTH QUARTER 2013. Article (CrossRef Link).

[6] Markopoulou A, Iannaccone G, Bhattacharyya S, Chuah C N, and Diot C, “Characterization of Failures in an Operational IP Backbone Network,” *IEEE/ACM Transactions on Networking*. Vol. 16, no. 4 pp. 749 - 762, Aug. 2008. Article (CrossRef Link).

[7] Gill P, Jain N, and Nagappan N, "Understanding network failures in data centers: measurement, analysis, and implications," in *Proc. of ACM SIGCOMM 2011*, pp. 350-361, Aug 15-19. 2011. Article (CrossRef Link).

[8] Yu H, An and V , Qiao C and Sun G , “Cost Efficient Design of Survivable Virtual Infrastructure to Recover from Facility Node Failures,” in *Proc. of IEEE International Conference on Communications (ICC)*, pp. 1-6, June 5-9, 2011. Article (CrossRef Link).

[9] Guo T, Wang N, Moessner K and Tafazolli R , “Shared Backup Network Provision for Virtual Network Embedding,” in *Proc. of IEEE International Conference on Communications (ICC)*, pp. 1-5, June 5-9, 2011. Article (CrossRef Link).

[10] Rahman M, Aib I and Boutaba R, “SVNE: Survivable Virtual Network Embedding Algorithms for Network Virtualization,” *IEEE Transactions on Network and Service Management (TNSM)*, vol. 10, no. 2, pp. 105-118, JUNE 2013. Article (CrossRef Link).

[11] Zhu Y and Ammar M, “Algorithms for assigning substrate network resources to virtual network components” in *Proc. of IEEE INFOCOM* , pp. 1-12, Apr. 23-29, 2006. Article (CrossRef Link).

[12] Fan J and Ammar M, “Dynamic topology configuration in service overlay networks - a study of reconfiguration policies,” in *Proc. of IEEE INFOCOM*, pp. 1-12, Apr. 23-29, 2006. Article (CrossRef Link).

[13] Yu M, Yi Y, Rexford J, and Chiang M , “Rethinking virtual network embedding: Substrate support for path splitting and migration,” *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 19- 29, APR. 2008. Article (CrossRef Link).

[14] Chowdhury N, Rahman M, and Boutaba R , “Virtual network embedding with coordinated node and link mapping,” in *Proc. of IEEE INFOCOM*, pp. 783–791, Apr. 19-25, 2009. Article (CrossRef Link).

[15] Shihabur Chowdhury, Reaz Ahmed, MD MASHRUR ALAM KHAN, et al, “Dedicated Protection for Survivable Virtual Network Embedding,” *IEEE Transactions on Network and Service Management (TNSM)*, vol. 10, no. 2, pp. 105-118, 2016. Article (CrossRef Link).

[16] H. Yu, et al, ”Regional Failure-Resilient Virtual Infrastructure Mapping in a Federated Computing and Networking System,” *Journal of Optical Communications and Networking*, Vol. 6, Issue 11, pp. 997-1007, 2014. Article (CrossRef Link).

[17] Liu X, Wang Y, Xiao A, et al., “Disaster-prediction based virtual network mapping against multiple regional failures,” in *Proc. of 2015 IFIP/IEEE International Symposium on* Integrated Network Management (IM), pp. 371-378, May 11-15, 2015. Article (CrossRef Link).

[18] Cai Z , Liu F , Xiao N , Liu Q and Wang Z, “Virtual Network Embedding For Evolving Networks,” in *Proc. of IEEE Global Telecommunications Conference (GLOBECOM)*, pp. 1-5, Dec. 6-10, 2010. Article (CrossRef Link).

[19] Liao D, Sun G, Anand V, et al., “Efficient provisioning for multicast virtual network under single regional failure in cloud-based datacenters,” *KSII TIIS* , vol. 7, no. 7, pp. 2325-2349, Jul 2014. Article (CrossRef Link).

[20] Ayoubi S, Chen Y, Assi C, et al., “Multicast tree repair and maintenance in the cloud,” in *Proc. of 8th IEEE Int. Conf. on Cloud Computing*, pp. 829 - 835, June 27 - July 2, 2015. Article (CrossRef Link).

[21] Zegura E W, Calvert K L, and Bhattacharjee S, “How to model an internetwork,” in *Proc. of IEEE INFOCOM’96*, pp. 594-602, March24-28, 1996. Article (CrossRef Link).

[22] “GNU Linear Programming Kit,” http://www.gnu.org/software/glpk/.

Ying Wang received the Ph.D. degree from Beijing University of Posts and Telecommunications in 2006. She is currently an associate professor in the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. Her research interests include the management of future network, cloud computing and IT service management.

Qingyun Chen is currently a M.S. candidate at Beijing University of Posts and Telecommunications. His research interest focuses on future network management.

Wenjing Li is currently a professor in the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. Her research interests include the management of future network, wireless network and automatic management.

Xuesong Qiu received the Ph.D. degree from Beijing University of Posts and Telecommunications in 2000. He is currently a professor in the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. His major research interests include network management and telecommunication software.