

Research Article

A Multidomain Survivable Virtual Network Mapping Algorithm

Xiancui Xiao,^{1,2} Xiangwei Zheng,^{1,2} and Yuang Zhang^{1,2}

¹School of Information Science and Engineering, Shandong Normal University, Jinan 250014, China

²Shandong Provincial Key Laboratory for Distributed Computer Software Novel Technology, Jinan 250014, China

Correspondence should be addressed to Xiangwei Zheng; xwzhengcn@163.com

Received 18 August 2017; Accepted 11 October 2017; Published 13 November 2017

Academic Editor: Lianyong Qi

Copyright © 2017 Xiancui Xiao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Although the existing networks are more often deployed in the multidomain environment, most of existing researches focus on single-domain networks and there are no appropriate solutions for the multidomain virtual network mapping problem. In fact, most studies assume that the underlying network can operate without any interruption. However, physical networks cannot ensure the normal provision of network services for external reasons and traditional single-domain networks have difficulties to meet user needs, especially for the high security requirements of the network transmission. In order to solve the above problems, this paper proposes a survivable virtual network mapping algorithm (IntD-GRC-SVNE) that implements multidomain mapping in network virtualization. IntD-GRC-SVNE maps the virtual communication networks onto different domain networks and provides backup resources for virtual links which improve the survivability of the special networks. Simulation results show that IntD-GRC-SVNE can not only improve the survivability of multidomain communications network but also render the network load more balanced and greatly improve the network acceptance rate due to employment of GRC (global resource capacity).

1. Introduction

Network virtualization enables multiple virtual networks (VNs) to coexist on the same physical network dynamically, so that virtual network users can share the underlying physical network [1]. At the same time, network virtualization technology as a new technology means to provide a solution to cloud computing diverse services [2, 3]. In order to relieve the interdependence between network control and data plane [4, 5], network service operators are generally divided into two roles: the underlying infrastructure provider (InP) and the service provider (SP); their tasks are to deploy the underlying network resources and lease a number of underlying infrastructure providers to provide the underlying network resources to meet the custom scalability of virtual network services [6, 7].

This paper refers to the relevant algorithms in the literature [8–12] and combines the multidomain mapping problem with the network survivability. We classify process as two stages, the resource classification process and the network mapping process, and the resource classification process is further divided into two parts, the primary flow

and the backup flow resource, which can, respectively, form a complete underlying physical network topology. The characteristic of this paper is that we protect the security of node data by mapping the virtual nodes to different domains of the physical network. Simultaneously, in order to ensure the balance of network load, we choose a new measure, global resource capacity (GRC) [13], which can measure the potential mapping capability of nodes in the process of node mapping. IntD-GRC-SVNE is applied in a multidomain environment and when compared with the traditional algorithm of IntD-GREEDY-SVNE and random algorithm IntD-RANDOM-SVNE, the acceptance rate, network load balancing, and network revenue of IntD-GRC-SVNE all show better performance. In addition, the non-cross-allocation of the primary flow and backup flow resources in this paper can greatly improve the survivability of the network.

This paper is organized as follows: we review related work in Section 2. We introduce problem description in Section 3 and provide survivable virtual network mapping model in Section 4. The details of virtual network mapping algorithm are given in Section 5. Finally, we analyze experiments and provide discussion in Section 6.

2. Related Work

As the core of network virtualization technology, the goal of virtual network mapping is to provide node and link resources for dynamic virtual requests to meet their mapping service requirements. However, with the number of network users climbing, the user requirements for the stability of the network are getting higher and higher. In order to improve the survivability of the network, many researchers have done some researches and most of the current researches focused on both protection and backup [14].

Physical failure is generally divided into node failure and link failure. In order to deal with physical link failure, Rahman et al. [15] designed a SVNE method for passive allocation of backup resources, which is essentially a virtual link remapping method after a failure. However, because the node failure will inevitably lead to its adjacent link failure, it is necessary to rebuild the virtual nodes affected by the virtual node and the virtual nodes affected by the virtual link, so the recovery of node failure is relatively complex. Guo et al. [16] proposed a SVNE method for proactively allocating backup resources for virtual links in the initial mapping of VNs which can be timely recovery network failure. By this way, the economic losses caused by network failures can be minimized in the case of efficient use of resources.

In order to guarantee the survivability of VN under the influence of physical node failure, Yu et al. [17] designed method of reliable VN two-stage SVNE method; they firstly extended VN to a reliable network that backs up important virtual nodes by adding redundant nodes and corresponding redundant links to the original VN and secondly mapped the reliable VN to the physical network with the goal of maximizing shared backup link resources. Qiao et al. [18] used another two-stage SVNE method to further reduce the backup overhead, but, after a node failure, the method may need to migrate a large number of working virtual nodes in addition to migrating affected virtual nodes. Hu et al. [14] defined and solved the SVNE problem of the constraint position, emphasizing that the backup node used for remapping must satisfy the position constraint specified by Yeow et al. [19] who used the SVNE method of ORP (opportunistic redundancy pooling) to dynamically create and pool backup virtual nodes and made different VNs share backup virtual nodes for better fault tolerance and cost performance.

The number of researches in the literature on the protection of node data is still relatively small, especially in the event of a major natural disaster such as earthquakes and tsunamis which may lead to domain network failure and serious loss of node data in a moment [20–22]. In this situation, the existing node protection mechanism will lose its role. Therefore, implementing multidomain mapping can not only protect the independence of the various data sets but also avoid domain data loss caused by the failure. At this stage, there are a few studies; Papagianni et al. [9] used the advantages and related principles of cloud computing to achieve the distribution of files between multiple geographies and the process of document distribution for the cloud computing in this paper is actually the realization of a

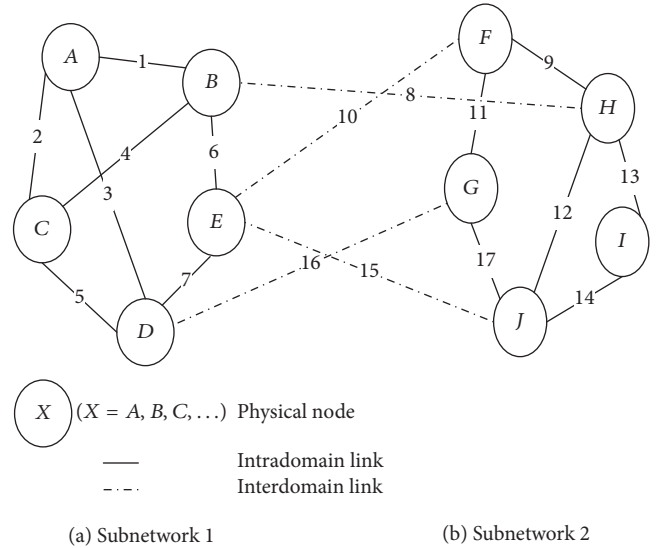


FIGURE 1: Physical network.

multiarea information exchange. Mijumbi et al. [8] also put forward using multidomain advantages to achieve resource allocation in the virtual network environment. Nonetheless, there are few existing literatures that really solve the problem of network mapping.

3. Problem Description

In this section, we first give a multidomain underlying physical network, virtual request network model, and their formal descriptions. Then we give a description of survivable virtual network mapping algorithm that supports multidomain mapping.

3.1. Physical Network. The underlying network topology is marked as a weighted undirected graph $G_s = (N_s, L_s, C_s^n, C_s^l)$, where N_s represents a collection of underlying network nodes, L_s represents a collection of underlying network links, C_s^n represents the computational power of the physical node, and C_s^l indicates the link bandwidth corresponding to the physical path. Figure 1 depicts an example of an underlying physical network which includes two subnetworks in different domains. Tables 1 and 2 show the geographical coordinates of the physical nodes in the two sub-networks, respectively represents the relative position between the physical network nodes and the available computing resources of the nodes CPU. Table 3 shows the available bandwidth resources of physical links.

As shown in Figure 1, circles filled with different English alphabets such as A, B, and C and so on represent different physical nodes. The physical node is a physical, active electronic device attached to the substrate network and can send, receive, or forward information via communication channel.

There are two types of physical links: the interdomain link which is used to complete the interdomain communication and the intradomain link in a single-domain network which

TABLE 1: Subnetwork 1 node information.

Physical node	Coordinate position (x, y)	Computing resources, CPU
A	(57, 54)	57
B	(36, 46)	39
C	(47, 60)	34
D	(40, 11)	24
E	(37, 33)	25

TABLE 2: Subnetwork 2 node information.

Physical node	Coordinate position (x, y)	Computing resources, CPU
F	(67, 74)	57
G	(76, 66)	39
H	(65, 60)	34
I	(70, 81)	24
J	(77, 63)	25

is used to complete intradomain communication. No matter what kinds of underlying links are there and how many virtual links can be mapped, the substrate connection is a physical communication channel between two substrates nodes.

3.2. Virtual Network. The undirected graph of the virtual network request is similar to the undirected graph of the underlying network. The network topology map is marked as a weighted undirected graph $G_V = (N_V, L_V, R_V^n, R_V^l)$, where N_V represents a collection of virtual network nodes, L_V represents a collection of virtual network links, R_V^n represents the computing power requirements of the virtual node, and R_V^l indicates the bandwidth resource requirements of the virtual link. Figure 2 depicts an example of a virtual request with node constraints and link constraints. Table 4 shows the relative geographical coordinates of the virtual nodes, node computing resource, CPU, and maximum mapping distance which is defined as formula (1), and Table 5 shows the bandwidth resource requirement of the virtual links in a virtual request.

$$\text{Lim_Dis} \left(\text{Lim_Dis} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \right). \quad (1)$$

As shown in Figure 2, hexagons filled with different English alphabets a , b , and c represent different virtual nodes. A virtual node is a software component that has a host or routing function, such as an operating system encapsulated in a virtual machine. Virtual nodes form a virtual network topology through the virtual link interconnection.

The lines identified as 1 and 2 are virtual links which are logical interconnections between virtual nodes. For a virtual network, the function of virtual link is to connect directly with the physical network and dynamically display the user's resource requirements.

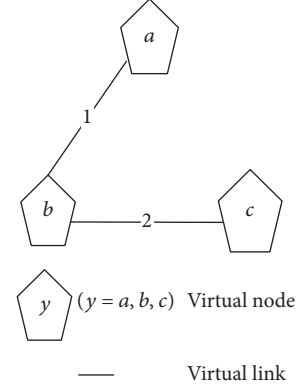


FIGURE 2: Virtual request.

3.3. Global Resource Capacity. In the node mapping phase, the corresponding mapping nodes need to be found on the underlying physical network for each virtual node. In general, all physical nodes with more resources than the resources required by the virtual node can be used as candidate nodes for virtual nodes. In this paper, we obtain the unit of measurement $GRC(N_i)$ which is used to measure the potential mapping ability of all the physical nodes through the global resource calculation method in the process of node selection. The calculation method is defined as follows:

$$\begin{aligned} GRC(N_i) &= (1 - d) C_r(N_i) \\ &+ d \sum_{v \in \text{Neb}(N_i)} \frac{\text{bw}(i, j)}{\sum_{k \in \text{Neb}(N_j)} \text{bw}(k, j)} GRC(N_j). \end{aligned} \quad (2)$$

In (2), d is set to a constant representation of the decreasing factor, $\text{bw}(i, j)$, $(i, j) \in E$, represents link resources, and $\text{Neb}(N_i)$ represents a set of physical nodes adjacent to the node N_i . In addition, $C_r(N_i)$ represents the proportion of the node computing resources that occupy the entire network computing resource. The calculation formula is defined as follows:

$$C_r(N_i) = \frac{\text{CPU}(N_i)}{\sum_{N_v \in V} \text{CPU}(N_v)}, \quad \forall N_i \in V. \quad (3)$$

The calculation of GRC_S for all nodes using the vector format can be defined as follows:

$$GRC = (1 - d) C_R + d M_R GRC, \quad (4)$$

where $GRC = (GRC(N_1), GRC(N_2), GRC(N_3), \dots, GRC(N_{|V|}))^T$, $C_R = (C_{r_1}, C_{r_2}, C_{r_3}, \dots, C_{r_{|V|}})^T$, and M_R is a matrix of $|V| * |V|$ dimensions. Each dimension data corresponds to two adjacent contacts $m(u, v)$ value which is defined as follows:

$$m(u, v) = \begin{cases} \frac{\text{bw}(u, v)}{\sum_{x \in \text{Neb}(v)} \text{bw}(x, v)} & (u, v) \in E \\ 0, & \text{other.} \end{cases} \quad (5)$$

TABLE 3: Link resource information.

Link serial number	Link available bandwidth resources (bw)	Interdomain communication link (Y: 1, N: 0)
1	74.827859	0
2	83.370701	0
3	77.140495	0
4	95.433926	0
5	66.452354	0
7	69.448338	0
8	56.235293	1
9	57.188120	0
10	77.287520	1
11	54.428015	0
12	96.737798	0
13	88.669225	0
14	81.164182	0
15	60.910143	1
16	51.048862	1
17	79.065118	0

TABLE 4: Virtual network node information.

Virtual node	Coordinate position (x, y)	Resources required, CPU	Distance limit (Lim_Dis)
a	(55, 50)	10	75
b	(37, 40)	35	60
c	(80, 96)	20	40

TABLE 5: Resource requirements of virtual links.

Link serial number	Link bandwidth resources required (bw)
1	74.827859
2	83.370701
3	77.140495

3.4. *Evaluation Indices.* The primary evaluation index of the network is defined as follows:

(1) *Network Acceptance Rate*

$$\frac{\lim_{T \rightarrow \infty} \sum_{t=0}^T \text{VNR}}{\sum_{t=0}^T \text{VNR}_S}. \quad (6)$$

In (6), $\sum_{t=0}^T \text{VNR}$ represents the number of virtual networks successfully mapped from $t = 0$ to T and $\sum_{t=0}^T \text{VNR}_S$ represents the total number of virtual network requests from $t = 0$ to T .

(2) *Average Cost of the Network*

$$\frac{\lim_{T \rightarrow \infty} \sum_{t=0}^T \text{Cost}(G_V, t)}{\sum_{t=0}^T \text{VNR}_S}. \quad (7)$$

In (7), $\sum_{t=0}^T \text{Cost}(G_V, t)$ represents the resources required to successfully map the virtual network from $t = 0$ to T .

(3) *Node Pressure*

$$\text{Node_Load}(n^s) = \sum_{v \in E^v} \frac{\text{Map}_{n^s}(C_n^v)}{C_n^s}. \quad (8)$$

In (8), C_n^s indicates the node resources of the physical node n^s and $\text{Map}_{n^s}(C_n^v)$ represents the sum of the resources including all virtual nodes mapped on physical node n^s .

(4) *Link Pressure*

$$\begin{aligned} & \text{Link_Load}(l^s) \\ &= \sum_{v \in E^v, p \in (p, v)} \frac{\text{Map}_l [C_{l_1}^v(p, v) + C_{l_2}^v(p, v)]}{C_l^s}. \end{aligned} \quad (9)$$

Link pressure is divided into two parts, the primary flow link pressure and backup flow link pressure. As the former is used to provide link resources under the normal service of the network, the latter is used to provide backup link resources for network failure. In (9), C_l^s indicates a physical link l and $\text{Map}_l [C_{l_1}^v(p, v) + C_{l_2}^v(p, v)]$ represents the total resources consumption including primary and backup resources.

(5) *Long-Term Network Revenue*

$$\frac{\lim_{T \rightarrow \infty} \sum_{t=0}^T \text{Rev}(G_V, t)}{T}. \quad (10)$$

In (10), $\sum_{t=0}^T \text{Rev}(G_V, t)$ represents the total revenue obtained from the virtual network that is successfully mapped from $t = 0$ to T .

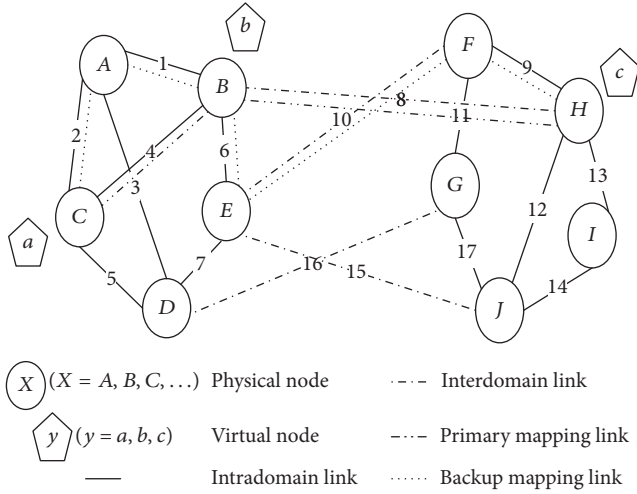


FIGURE 3: Network mapping.

4. Survivable VN Mapping Model for Multidomain Network

4.1. Mapping Problem Description. Virtual network mapping problems are generally defined as mapping: $M \in G_v(N_v, L_v) \rightarrow G_s(N_s, L_s)$ which usually include node mapping and link mapping. The number of physical network resources is deemed to be shown in Figure 1. When the virtual request arrives, in the node mapping phase, the candidate physical node satisfies two conditions: (1) the number of available computing resources is greater than the amount of CPU resources required by the virtual node and (2) the physical coordinates of the physical node meet the requirements of the virtual node.

As shown in Figure 3, the virtual nodes a and b are, respectively, mapped on the physical nodes B and C in subnetwork 1 according to the mapping requirements and the node selection method GRC. But the virtual node c cannot find the mapping node in subnetwork 1 under the limit of the maximum mapping distance. Based on previous studies, under the single-domain network provider environment, the virtual request in this paper will fail to map.

In order to minimize the geographical coordinates caused by the failed mapping and improve the acceptance rate of the network, the proposed IntD-GRC-SVNE in this paper supports multidomain network mapping, so we can find a feasible physical node in the network outside subnetwork 1. For example, as shown in Figure 3, the physical node H is found as the mapping node in subnetwork 2.

4.2. Data Security. As the nodes are mapped in different domain subnetworks, when a major accident happened in a subnetwork, such as earthquakes, floods, or other large natural disasters, mapping in different domain subnetworks can effectively avoid the loss of data and improve the security of the data.

As shown in Figure 3, for example, when a major natural disaster occurred at the location of the domain, subnetwork 2

will result in paralyzed service process and large loss of data. In accordance with the previous mapping, the virtual request mapped to subnetwork 2 will be interrupted. In this paper, the virtual request is mapped to different domain subnetworks; in the case of the paralyzed network, the mapping method of IntD-GRC-SVNE can not only ensure that some nodes and links work properly but also improve the node data security to a certain degree.

At the same time, in order to prevent the impact of link failure, IntD-GRC-SVNE sets the primary resources and backup resources and the proportion of the former is $\alpha(s)b(s)$ and the proportion of the latter is $\beta(s)b(s)$. These two types of resources cannot be crossed in order to ensure the effective use of resources; that is, $\alpha(s) + \beta(s) = 1$.

The network mapping process is shown in Figure 3, virtual link 1 is mapped to the primary flow path which is identified as $(a, b) \rightarrow (C, B)$, and the backup flow path is defined as $(a, b) \rightarrow (C, A, B)$. Virtual link 2 is mapped to the primary flow path which is identified as $(b, c) \rightarrow (B, H)$ and the backup flow path is defined as $(b, c) \rightarrow (B, E, F, H)$. Once the virtual request is accepted, the resources on the primary flow resource path are always used until the virtual request service ends, so the primary flow resource does not support resource sharing. The backup flow path is used to ensure that the network can quickly reroute and make the network reprovide normal service in the event of a failure.

4.3. Business Utility Model for Multidomain Mapping. Based on the description of the network mapping problem in Section 4.1, we give the function model of the problem in this section. During the virtual network mapping process, the cost of the network mainly includes two parts, the node cost and the link cost, and the latter includes intradomain link cost and interdomain link cost. The two costs are defined in different ways: the intradomain link cost is defined as (11) and interdomain link cost is defined as (12):

$$\begin{aligned} \text{Inter_Cost}(G^v) &= \sum_{v \in E^v} \text{CPU}(n^v) \\ &+ \sum_{v \in E^v, p \in p(v)} \omega [b_1(p, v) + b_2(p, v)], \end{aligned} \quad (11)$$

$$\begin{aligned} \text{Dom_Cost}(G^v) &= \sum_{v \in E^v, p \in p(v)} \text{Dom}_.b_1(p, v) + \text{Dom}_.b_2(p, v). \end{aligned} \quad (12)$$

In (11), n^v represents any virtual node in a virtual request and $\text{CPU}(n^v)$ represents the computing resource occupied by the virtual node n^v . In (11) and (12), $b_1(p, v)$ and $\text{Dom}_.b_1(p, v)$, respectively, represent the primary flow path within the intradomain and interdomain resource consumption, while $b_2(p, v)$ and $\text{Dom}_.b_2(p, v)$, respectively, represent the backup flow path within the intradomain and interdomain resource consumption. As ω represents the balance parameter between the node cost and the link cost, $\sum_{v \in E^v} \text{CPU}(n^v)$ represents the consumption of node

resources. While $\sum_{v \in E^v, p \in p(v)} \omega[b_1(p, v) + b_2(p, v)]$ represents primary flow of interdomain resource consumption, $\text{Dom}.b_1(p, v) + \text{Dom}.b_2(p, v)$ represents backup flow of interdomain resource consumption. So the total cost is defined as follows:

$$\text{TCO}(G^v) = \text{Inter_Cost} + \varphi \text{Dom_Cost}. \quad (13)$$

The goal of the algorithm in this paper is to achieve greater revenue by improving the acceptance rate of the network as much as possible in the case of satisfying the requirement of users. The relevant constraints of this algorithm are as follows (see (14)~(19)).

Resource capacity limits of primary flows and backup flows of intradomain physical link and interdomain physical link are as follows:

$$\begin{aligned} \sum_{v \in E^v, p \in p(v)} \vartheta_s(p) \{b_1(p, v), \text{Dom}.b_1(p, v)\} &\leq \mathfrak{R}_\alpha(s), \\ \forall s \in E^s, \\ \sum_{v \in E^v, p \in p(v)} \vartheta_s(p) \{b_2(p, v), \text{Dom}.b_2(p, v)\} &\leq \mathfrak{R}_\beta(s), \\ \forall s \in E^s. \end{aligned} \quad (14)$$

Resource capacity limits of primary flow and backup flow of virtual network link are as follows:

$$b(v) = \sum_{p \in p(v)} \{b_1(p, v), \text{Dom}.b_1(p, v)\}, \quad \forall v \in E^v, \quad (15)$$

$$b(v) \leq \sum_{p \in p(v)} \{b_2(p, v), \text{Dom}.b_2(p, v)\}, \quad \forall v \in E^v. \quad (16)$$

In (16), as the remaining resources of the primary flow are marked as $\mathfrak{R}_\alpha(s)$, the remaining resources of the backup flow are marked as $\mathfrak{R}_\beta(s)$ and $\vartheta_s(p)$ is a flag variable indicating whether the link is occupied; when the value is 1, $s \in p$; otherwise $s \notin p$.

The primary and backup flow resource cannot be crossed:

$$\begin{aligned} \vartheta_s(p) \vartheta_s(q) [\delta_1(p, v) + \delta_2(p, v)] &\leq 1, \quad \forall s \in E^s \\ \{b_1(p, v), \text{Dom}.b_1(p, v)\} &\leq b(v) \delta_1(p, v), \\ \forall p \in p(v) \quad (17) \\ \{b_2(p, v), \text{Dom}.b_2(p, v)\} &\leq b(v) \delta_2(p, v), \\ \forall p \in p(v). \end{aligned}$$

Range of variables is as follows:

$$\delta_1(p, v) \in \{0, 1\}, \quad \forall v \in E^v, \forall p \in P(v), \quad (18)$$

$$\delta_2(p, v) \in \{0, 1\}, \quad \forall v \in E^v, \forall p \in P(v). \quad (19)$$

In (18)~(19), δ_i is a flag used to represent whether the primary resource and backup resource are cross-occupied. The initial value of δ_i is set to 0; once the resources are occupied during the mapping process, δ_i will be set to 1.

5. Virtual Network Mapping Algorithm (IntD-GRC-SVNE)

5.1. Flow Chart of IntD-GRC-SVNE. The implementation of this algorithm is in accordance with the order of arrival of the events. At different time points, the algorithm will deal with different events which include two types of virtual network requests: the new virtual request waits for service and the virtual request leaves after service has been completed. The algorithm flow will not be completed until all the events have been processed.

In Figure 4, the core part of this algorithm is to find the mapping scheme in the multidomain environment and the key is to find the physical mapping nodes distributed in different domain subnetworks for virtual nodes on one link. Through the above mapping method, we can improve the node data security and avoid data loss due to domain failures and the process of finding a mapping scheme is illustrated in Figure 5.

5.2. Algorithmic Description of IntD-GRC-SVNE. The survivable virtual network mapping algorithm (GRC-SVNE) is described in Algorithm 1.

6. Experiments and Discussions

6.1. Experimental Settings. In this paper, the topology and location information of the network are randomly generated by the GT-ITM tools. The underlying network topology which consists of six domain subnetworks includes 100 nodes and 570 links. The node CPU resource and bandwidth resource in each domain subnetwork obey a uniform distribution of 50–100. The rest of the parameters involved in the paper are shown in Table 6. The numbers of nodes and intradomain links are shown in Table 7, while the number of interdomain links is shown in Table 8.

We assume that the number of virtual network requests arriving in 100 time-units obeys the *Poisson* process with an average of 5 and the lifetime of each virtual network is also exponentially distributed with an average lifetime of 500 time-units. For any virtual network request, the number of network nodes is uniformly distributed between 2 and 20, and any two virtual network nodes are connected with a probability of 0.8. The numbers of virtual network node resources and link bandwidth resource requirements are uniformly distributed between 0 and 50. The coordinates of the nodes x and y variables are uniformly distributed from 0 to 100, assuming that the location constraints D of all virtual network mapping requests are constants. Each simulation experiment runs about 10,000 time-units and contains 100 virtual network requests.

6.2. Results and Discussions. As IntD-GRC-SVNE supports multidomain virtual network mapping, in order to improve the network load balancing, the algorithm uses the existing metric GRC that measures the potential mapping capability of physical nodes to select the more reasonable nodes as the mapping nodes in node mapping phase. In addition, in order to demonstrate the performance of the algorithms

TABLE 6: Simulation parameters.

Node number of the substrate network	100
Link number of the substrate network	570
Initial available computing resources on substrate nodes	50–100 units
Initial available bandwidth resources on substrate links	50–100 units
Average lifetime of the VNRs	500 time-units
Bandwidth demand of a virtual link	0–50 units
Computing resource demand of a virtual node	0–50 units
Node number in a VNR	10–25
$\alpha(s)$	0.5
$\beta(s)$	0.5

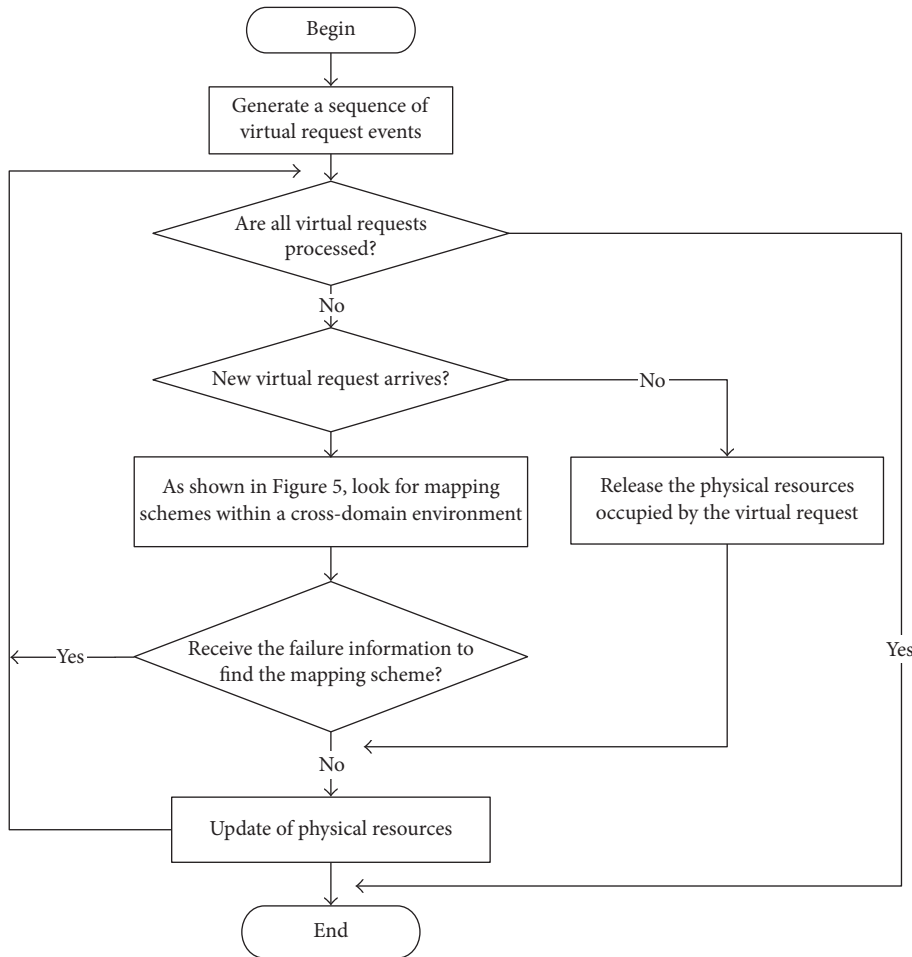


FIGURE 4: Flow chart of network mapping.

in multidomain networks, this paper applies the traditional random algorithm (named as IntD-RANDOM-SVNE in experimental results) and greedy algorithm (named as IntD-GREEDY-SVNE in experimental results) to multidomain network environment. The acceptance rate, average cost, and long-term revenue of the experimental comparison results between the three algorithms are shown in Figures 6–8.

As shown in Figure 6, IntD-GRAND-SVNE and IntD-GREEDY-SVNE are applied in the multidomain environment and then compared with the algorithm proposed

in this paper. Contrast simulation experiments show that IntD-GRC-SVNE reflects obvious advantages in the aspect of acceptance rate. IntD-RANDOM-SVNE, IntD-GREEDY-SVNE, and IntD-GRC-SVNE are all applied in the multidomain environment in this paper, and when compared with the traditional algorithm, IntD-GRC-SVNE proposed in this paper is more survivable. In addition, because IntD-GRC-SVNE uses a new node selection scheme (using a new metric GRC to measure the potential mapping capabilities of physical nodes), the process of selecting nodes is more

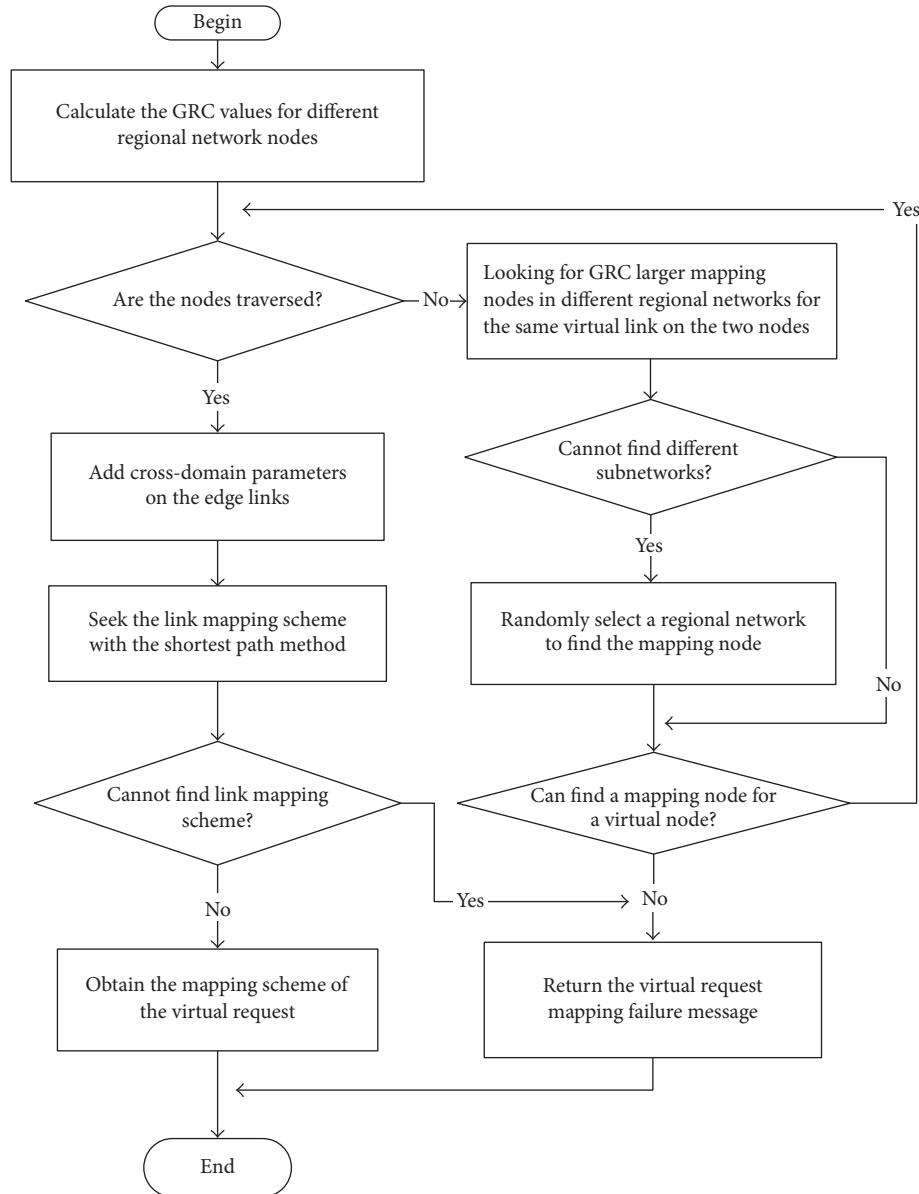


FIGURE 5: Flow chart of searching a mapping scheme in multidomain environment.

TABLE 7: Resource information for each domain subnetwork.

Domain network serial number	Number of nodes	Number of links
0	19	38
1	16	34
2	15	27
3	17	32
4	18	38
5	15	26

reasonable, which further makes the utilization of node resources more efficient and reduces node mapping failure rate. Therefore, the mapping success rate of IntD-GRC-SVNE

is significantly higher than the other two traditional algorithms IntD-RANDOM-SVNE and IntD-GREEDY-SVNE.

In addition, the algorithms used in this paper preferentially look for mapping nodes in different network domains. So IntD-RANDOM-SVNE, IntD-GREEDY-SVNE, and IntD-GRC-SVNE have higher data security guarantees than those algorithms previously mapped in a single-domain network environment.

As shown in Figure 7, at the beginning of the runtime, the cost of IntD-GREEDY-SVNE is much lower than IntD-RANDOM-SVNE and the IntD-GRC-SVNE proposed in this paper. With the passage of service time, due to the unreasonable use of resources, the available resource in IntD-GREEDY-SVNE is not enough and further causes network costs to rise.


```

(1) Create a number of virtual network request events
(2) The physical link resources are divided into the primary and backup flow resources, and set the ratio as 1:1.
(3) WHILE (Virtual network events are not fully processed)
(4)   IF (The event type is a new virtual network request)
(5)     Calculate the GRC value of the physical node according to formula (1)
(6)     FOR (The number of virtual nodes)
(7)       FOR (The number of virtual links connected to the node)
(8)         The two nodes on the same virtual link are mapped in two different domain sub-networks.
(9)         IF (Cannot find a different sub-network)
(10)          Randomly select a domain network as a mapping network and put the candidate physical node
              number found in the candidate [i].
(11)         ELSE
(12)          Find the candidate physical node within the selected sub-network and place the serial number in
              the candidate [i].
(13)         ENDIF
(14)       ENDFOR
(15)     IF (!candidate [i])
(16)       Return Node mapping failed.
(17)     ELSE
(18)       Select the eligible physical node in the candidate [i] as the mapping node according to the value of
              GRC from large to small;
(19)     ENDIF
(20)   ENDFOR
(21)   Form a virtual request node mapping scheme;
(22) ENDFOR
(23) FOR (The number of virtual links)
(24)   Use the Dijkstra method to find the primary flow path and the backup flow path between Node_From
      and Node_To;
(25)   IF (!Primary flow path || !Backup flow path)
(26)     WHILE (Candidate [i].size > 0)
(27)       Randomly re-select the physical node in Candidate [i] as the mapping node for the virtual node;
(28)       Delete the mapping node number in Candidate [i];
(29)       Use the Dijkstra method to find the primary flow path and the backup flow path for the new
              node scheme;
(30)       IF (Candidate [i].size == -1)
(31)         Return Link mapping failed;
(32)       ENDWHILE
(33)     ENDIF
(34)   ENDFOR
(35)   Form a new link mapping scheme;
(36) ENIF
(37) ELSE (The event type is the virtual network request to leave)
(38)   Update the virtual network request information;
(39)   Update physical network resources (including node resources, primary flow resources, and backup link
      flow resources);
(40) ENDWHILE

```

ALGORITHM 1

In general, the network cost consists of two parts: the node resource and the link resource. In the context of this experiment, the link resource includes the primary flow resource and the backup flow resource. In the case of its highest acceptance rate, the network cost of IntD-GRC-SVNE is still lower compared with the other two algorithms; this can fully show the rationality of the mapping scheme. As IntD-GRC-SVNE can not only realize the multidomain mapping to meet the data security requirements but also make network costs lower when compared to the traditional algorithms IntD-GREEDY-SVNE and IntD-RANDOM-SVNE in the

same multidomain network environment, the above experimental results and analysis demonstrate the practicality of IntD-GRC-SVNE proposed in this paper.

Figure 8 shows the degree of node load balancing. In order to show the network load gap between different algorithms, this paper uses the common variance solution method to get the pressure distribution characteristics of the node. As shown in Figure 8, IntD-GRC-SVNE is more balanced than the IntD-RANDOM-SVNE and IntD-GREEDY-SVNE nodes. Since IntD-GRC-SVNE not only supports the security of the multidomain mapping to protect the

TABLE 8: Interdomain link number information.

Domain subnetworks i	Domain subnetworks j	Number of communication links
0	1	24
0	2	34
0	3	22
0	4	21
0	5	9
1	2	23
1	3	29
1	4	8
1	5	18
2	3	28
2	4	42
2	5	28
3	4	22
3	5	37
4	5	30

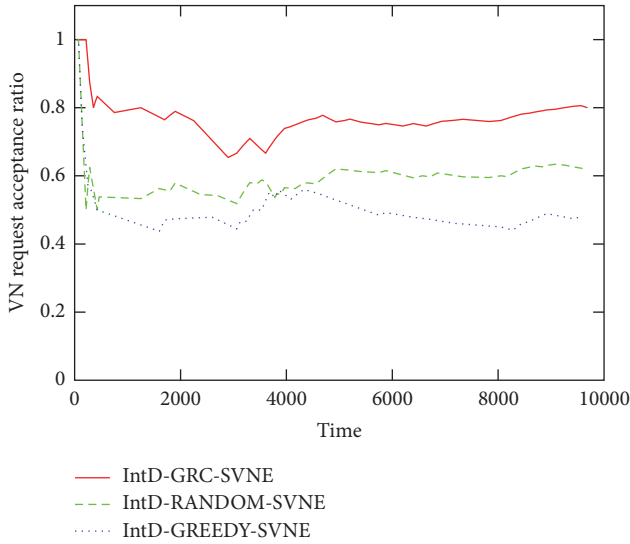


FIGURE 6: Acceptance rate.

node data but also applies the existing measurement metric GRC to the survivability network environment to make the network load more balanced, in this paper, all the required resources need to be found in a multidomain environment. The different amounts of resources in subnetwork can easily lead to network load imbalance, as GRC which is used in the paper indicates that the node mapping capabilities in whole network; this can just solve network balance problem. IntD-GRC-SVNE proposed in this paper shows the improvement effect of load balancing compared to IntD-RANDOM-SVNE and IntD-GREEDY-SVNE.

Similar to the node pressure variance, Figure 9 shows the link pressure variance which is calculated by the traditional mathematical method. As shown in Figure 9, compared with IntD-RANDOM-SVNE and IntD-GREEDY-SVNE, the link

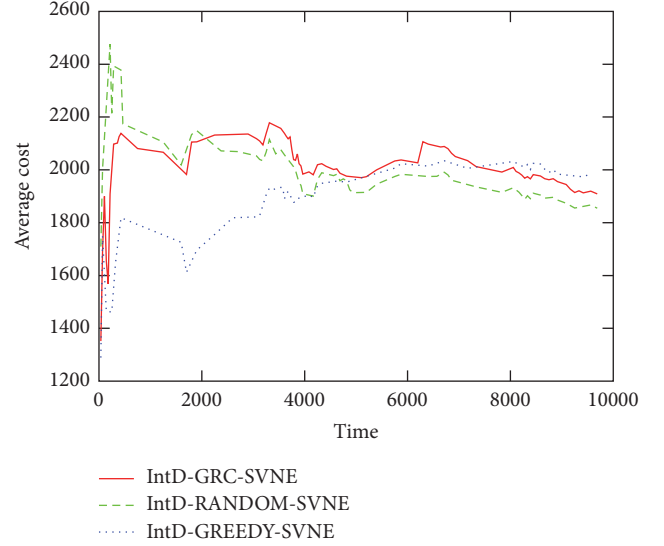


FIGURE 7: Network costs.

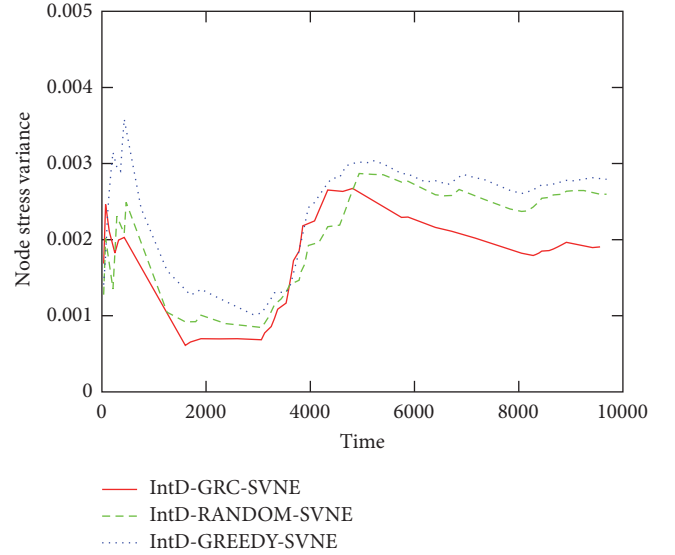


FIGURE 8: Node pressure variance.

load of IntD-GRC-SVNE is obviously more balanced which can also explain why the algorithm cost is low. In IntD-GRC-SVNE, the use of link resources is relatively balanced, so there are sufficient resources to receive more virtual requests. In IntD-GRC-SVNE, the GRC metric is applied to select the appropriate mapping node in the node selection process as the GRC value is determined by two kinds of resources: resources of the node itself and the number of link resources directly connected and indirectly connected to the node. The node selection process will affect the subsequent link mapping process to a certain extent. So IntD-GRC-SVNE selects the physical node which has better potential mapping capability as the mapping node. At the same time, it chooses a physical node with better potential mapping capability as the mapping node will reduce the difficulty of finding mapping links. Through the mapping, most of the selected mapping links can be better connected with other physical nodes,

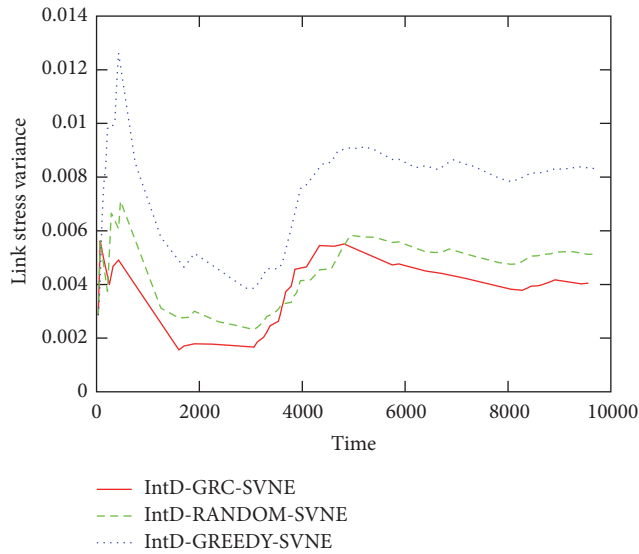


FIGURE 9: Link pressure variance.

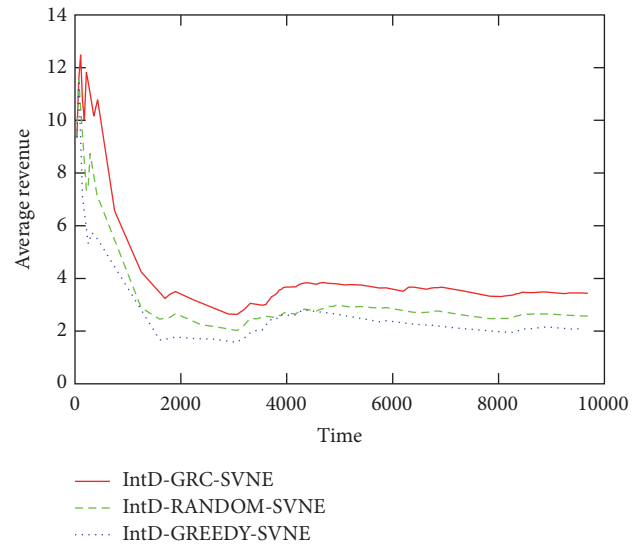


FIGURE 10: Average network revenue.

which not only improve the success rate of mapping but also improve the efficiency of the link and further make the link load more balanced.

There is also something that is worth emphasizing: because virtual requests are mapped across different domain networks, single-domain communication congestion is avoided. This also improves the load balancing of the network compared to the previous single-domain network mapping algorithm.

As shown in Figure 10, IntD-GRC-SVNE shows higher revenue when compared with the other two traditional algorithms, IntD-GREEDY-SVNE and IntD-RANDOM-SVNE. The average revenue from the network is derived from the associated costs of providing services for virtual requests. Therefore, to a certain extent, the amount of receiving virtual requests and the length of service time jointly determine the network provider revenue. As shown in Figure 6, IntD-GRC-SVNE has a higher receiving rate compared to the other algorithms and its corresponding network average revenue is also higher as shown in Figure 10.

In general, the mapping of virtual requests in different subnetworks can avoid the sudden termination of service due to domain disastrous paralysis of the network. Although the experimental results cannot directly reflect the algorithm to improve the security of the data as the three algorithms support multidomain network mapping to ensure the consistency of mapping conditions, in practical applications, there is no doubt that the virtual request mapping in multidomain network environment will improve data security.

7. Conclusion and Future Work

After summarizing the existing research on network survivability and node data security, this paper proposes a survivable virtual network mapping algorithm (IntD-GRC-SVNE) that supports multidomain network mapping. A new metric, GRC, is used to represent the potential mapping

capability of the node in the node selection phase. In order to facilitate the comparison of the experimental results, this paper has applied the existing greedy algorithm and random algorithm to the multidomain network environment, which are called IntD-RANDOM-SVNE and IntD-GREEDY-SVNE, respectively, in experimental results. The simulation results show that IntD-GRC-SVNE proposed in this paper not only realizes the multidomain mapping but also improves the survivability of the network. When compared with the two traditional algorithms, the acceptance rate, the network load, and the network revenue of IntD-GRC-SVNE embody the obvious advantages.

Further work mainly focuses on the following two aspects: (1) studying IntD-GRC-SVNE's adaptability in different physical network environments and the impact of physical resources on algorithm performance and (2) studying the multidomain network problem and attempting to reduce the impact of network services failure.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

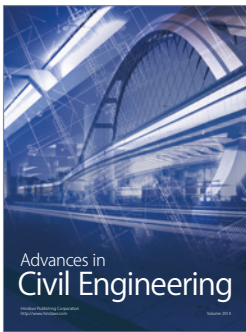
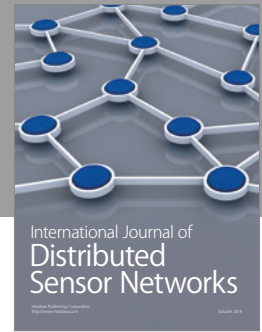
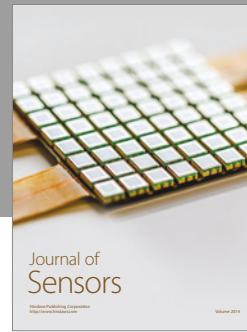
Acknowledgments

The authors are grateful for the support of the National Natural Science Foundation of China (61373149 and 61672329).

References

- [1] J. Carapinha and J. Jiménez, "Network virtualization," in *Proceedings of the the 1st ACM workshop*, p. 73, Barcelona, Spain, August 2009.
- [2] N. Feamster, L. Gao, and J. Rexford, "How to lease the internet in your spare time," pp. 61–64.
- [3] X.-W. Zheng, B. Hu, D.-J. Lu, Z.-H. Chen, and H. Liu, "Energy-efficient virtual network embedding in networks for cloud

- computing,” *International Journal of Web and Grid Services*, vol. 13, no. 1, pp. 75–93, 2017.
- [4] X. Cheng, Z. Zhang, S. Suet et al., “Review of Virtual Network Mapping Problem,” *Journal of Communications*, vol. 32, no. 10, pp. 143–151, 2011.
- [5] X. Cheng, S. Su, and Z. Zhang, “Virtual network embedding through topology-aware node ranking,” *SIGCOMM Computer Communication Review*, vol. 41, no. 2, pp. 38–47, 2011.
- [6] N. M. M. K. Chowdhury and R. Boutaba, “A survey of network virtualization,” *Computer Networks*, vol. 54, no. 5, pp. 862–876, 2010.
- [7] M. Chowdhury, M. R. Rahman, and R. Boutaba, “ViNEYard: virtual network embedding algorithms with coordinated node and link mapping,” *IEEE/ACM Transactions on Networking*, vol. 20, no. 1, pp. 206–219, 2012.
- [8] R. Mijumbi, J. Serrat, and J.-L. Gorricho, “Self-managed resources in network virtualisation environments,” in *Proceedings of the 14th IFIP/IEEE International Symposium on Integrated Network Management, IM 2015*, pp. 1099–1106, can, May 2015.
- [9] C. Papagianni, A. Leivadreas, and S. Papavassiliou, “A cloud-oriented content delivery network paradigm: modeling and assessment,” *IEEE Transactions on Dependable and Secure Computing*, vol. 10, no. 5, pp. 287–300, 2013.
- [10] X.-C. Xiao and X.-W. Zheng, “A proposal of survivable virtual network embedding algorithm,” *Journal of High Speed Networks*, vol. 22, no. 3, pp. 241–251, 2016.
- [11] G.-P. Gao, B. Hu, and J.-S. Zhang, “Design of a miniaturization printed circular-slot UWB antenna by the half-cutting method,” *IEEE Antennas and Wireless Propagation Letters*, vol. 12, pp. 567–570, 2013.
- [12] X. Yu, H. Wang, X. Zheng, and Y. Wang, “Effective algorithms for vertical mining probabilistic frequent patterns in uncertain mobile environments,” *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 23, no. 3-4, pp. 137–151, 2016.
- [13] J. Xu, J. Tang, K. Kwiat, W. Zhang, and G. Xue, “Survivable virtual infrastructure mapping in virtualized data centers,” in *Proceedings of the 2012 IEEE 5th International Conference on Cloud Computing (CLOUD ’12)*, pp. 196–203, June 2012.
- [14] Q. Hu, Y. Wang, and X. Cao, “Location-constrained survivable network virtualization,” in *Proceedings of the 35th IEEE Sarnoff Symposium (SARNOFF ’12)*, May 2012.
- [15] M. R. Rahman, I. Aib, and R. Boutaba, “Survivable virtual network embedding,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 6091, pp. 40–52, 2010.
- [16] T. Guo, N. Wang, K. Moessner, R. Tafazolli et al., “Shared backup network provision for virtual network embedding,” in *Proceedings of the 2011 IEEE International Conference on Communications (ICC ’11)*, pp. 1–5, June 2011.
- [17] H. Yu, V. Anand, C. Qiao, G. Sun et al., “Cost efficient design of survivable virtual infrastructure to recover from facility node failures,” in *Proceedings of the 2011 IEEE International Conference on Communications (ICC ’11)*, pp. 1–6, June 2011.
- [18] C. Qiao, B. Guo, S. Huang, J. Wang, T. Wang, and W. Gu, “A novel two-step approach to surviving facility failures,” in *Proceedings of the 2011 Optical Fiber Communication Conference and Exposition and the National Fiber Optic Engineers Conference (OFC/NFOEC ’11)*, pp. 1–3, March 2011.
- [19] W.-L. Yeow, C. Westphal, and U. C. Kozat, “Designing and embedding reliable virtual infrastructures,” *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 2, pp. 57–64, 2011.
- [20] W. Li, Y. Jiang, and J. Wen, “Risk Assessment of Industrial Space Network in Earthquake Disaster Scenarios-Taking Toyota Motor in Japan as an Example,” *Acta Geographica Sinica*, vol. 71, no. 8, pp. 1384–1399, 2016.
- [21] B. Cao, J. Li, and Q. Sun, “Power grid suffered earthquake damage response measures and optimization design , For electricity,” vol. 33, no. 3, pp. 76–82, 2016.
- [22] Z. Li and C. Jiang, “Study on Emergency Logistics Network Model for Facility Arming Considering Interruption Scenario,” *Periodical Issues*, vol. 30, 2016.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

