Hindawi Publishing Corporation
Discrete Dynamics in Nature and Society
Volume 2015, Article ID 316801, 8 pages
http://dx.doi.org/10.1155/2015/316801

Research Article

A Hybrid Reliable Heuristic Mapping Method Based on Survivable Virtual Networks for Network Virtualization

Qiang Zhu,[1] Hui-Qiang Wang,[1] Guang-Sheng Feng,[1] Hong-Wu Lv,[1] Zhen-Dong Wang,[2] Xiu-Xiu Wen,[1] and Wei Jiang[1]

[1] *College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China*
[2] *Faculty of Information Engineering, Jiangxi University of Science and Technology, Ganzhou 341000, China*

Correspondence should be addressed to Qiang Zhu; zhuqianghrbeu@163.com

Received 19 October 2014; Revised 8 December 2014; Accepted 16 December 2014

Academic Editor: Muhammad Naveed Iqbal

The reliable mapping of virtual networks is one of the hot issues in network virtualization researches. Unlike the traditional protection mechanisms based on redundancy and recovery mechanisms, we take the solution of the survivable virtual topology routing problem for reference to ensure that the rest of the mapped virtual networks keeps connected under a single node failure condition in the substrate network, which guarantees the completeness of the virtual network and continuity of services. In order to reduce the cost of the substrate network, a hybrid reliable heuristic mapping method based on survivable virtual networks (Hybrid-RHM-SVN) is proposed. In Hybrid-RHM-SVN, we formulate the reliable mapping problem as an integer linear program. Firstly, we calculate the primary-cut set of the virtual network subgraph where the failed node has been removed. Then, we use the ant colony optimization algorithm to achieve the approximate optimal mapping. The links in primary-cut set should select a substrate path that does not pass through the substrate node corresponding to the virtual node that has been removed first. The simulation results show that the acceptance rate of virtual networks, the average revenue of mapping, and the recovery rate of virtual networks are increased compared with the existing reliable mapping algorithms, respectively.

1. Introduction

Due to the “best effort” service model of the Internet, cloud computing is facing challenges in service diversity providing. Network virtualization is an effective technique to solve this problem [1–3]. Network virtualization allows multiple independent virtual networks cohabiting on a shared substrate network, which can quickly and cost-effectively carry out new types of business and technologies [4]. Virtual network requests consisting of virtual nodes and virtual links with resource constraints are generated by users. How to provide reasonable substrate network resources allocation for virtual network requests is called virtual network mapping, which has been proved to be a NP-hard problem [5, 6].

Previous researches on the virtual network mapping problem are almost concerned with the acceptance rate, the resource utilization, and the average revenue of the substrate network with no fault, such as ViNEYard algorithm [7], RW-BFS algorithm [8], DVNMA algorithm [9], distributed virtual network mapping algorithm [10], subgraph isomorphism detection based mapping algorithm [11], path splitting and migration [12], and other heuristic algorithms [13–15], without considering with the reliability of the mapped virtual networks. However, the nodes and the links in substrate network are inevitably affected by internal and external influences in actual situation, undermining the continuity of virtual network services.

To solve the reliable virtual network mapping problem, previous researches mainly use protection mechanisms [16]. Rahman and Boutaba [17] proposed a hybrid strategy heuristic algorithm by utilizing a prereserved quota for backing up on each substrate link. In order to reduce the redundant backup resources, Yu et al. [18] proposed a method that nodes and links are backed up in different areas for virtual networks, which requires virtual nodes to transmit the current states to the backup nodes and exacerbates substrate network resources consumption. Yeow et al. [19] proposed an optimizing redundancy pool to allocate backup resources to multiple virtual networks dynamically, which can reduce redundant resources consumption of the substrate network. The protection mechanisms preassign redundant substrate resources for virtual networks. The disadvantage of the protection mechanisms is that it will bring excessive consumption to the substrate network and impact acceptance rate of virtual networks and revenue of the substrate network in the long run. Unlike the protection mechanisms, Butt et al. [20] proposed a topology-awareness and reoptimization mechanism based on node migration and link remapping, which does not provide virtual networks with redundancies. If a node or link failure occurs, the affected virtual node migrates to normal substrate node and the virtual links are remapped in order to recover the virtual network. In addition, the literature [21] also uses a recovery mechanism to deal with the reliable virtual network mapping problem. The main problem of the recovery mechanisms is that the recovery rate will be low when there are few resources in the substrate network.

Inspired by the survivable routing in wavelength division multiplexing WDM (wavelength division multiplexing) optical network in literature [22], we propose the concept of survivable virtual network, which is different from traditional protection mechanisms and recovery mechanisms. In WDM optical networks, substrate resources are allocated to different logical networks. Any failure in the substrate network will lead to service interruptions. Survivable routing in WDM optical network means when the substrate network failure occurs, the affected optical network maintains the maximizing degree of connectivity, which can maintain a tolerable level of service and the ability for quick recovery [23, 24]. In this paper, the survivable virtual network needs to consider not only the links mapping, but also the nodes mapping, which is more complex than the survivable routing in WDM optical networks. To make the virtual networks with survivability, the mapping of virtual network needs a survivable strategy. When the substrate network is under a node failure, the rest of the mapped virtual networks needs to remain connected to ensure the continuity of virtual network services and finally uses the node migration and link remapping to repair the affected virtual networks.

In order to reduce the cost of the substrate network, a hybrid reliable heuristic mapping method based on survivable virtual networks is proposed. In Hybrid-RHM-SVN, we formulate the reliable mapping problem as an integer linear programming model. Firstly, we calculate the primary-cut set of the virtual network subgraph where the failed node has been removed. Then, we use ant colony optimization algorithm to achieve the approximate optimal mapping. The links in the primary-cut set should be selected from the substrate paths that do not pass through the substrate node corresponding to the virtual node that has been removed first.

2. Reliable Virtual Network Mapping Formulation

In this section, we provide a mathematical formulation for the reliable virtual network mapping problem. We map virtual networks with survivable strategy in the substrate network. The reliable virtual network mapping formulation is composed by the substrate network and virtual network requests modeling, virtual networks mapping problem modeling, and virtual network survivability analysis.

2.1. Substrate Network. Similar to the previous works in [7–21], the substrate network is modelled as an undirected graph by $G^s = (N^s, E^s, A_n^s, A_e^s)$, where N^s is the set of the substrate nodes and E^s is the set of the substrate links. A_n^s is an attribute set of each substrate node $n^s \in N^s$ including available CPU computing capability $\mathrm{cpu}(n^s)$ and location $\mathrm{loc}(n^s)$. A_e^s is an attribute set of each substrate link $e_{ij}^s \in E^s$ between i and j. The attribute of the substrate links is available bandwidth capability $b(e^s)$. A_n^s and A_e^s can be extended by adding other attributes such as memory capability and link delay. P^s is the set of loop-free paths in the substrate network, and $P^s(i, j)$ is the set of the loop-free path between i and j.

2.2. Virtual Network Request. Similar to the substrate network, a virtual network request is also modelled as an undirected graph by $G^v = (N^v, E^v, C_n^v, C_e^v)$, where N^v is the set of virtual nodes and E^v is the set of virtual links. C_n^v is the attributes set of virtual nodes including available CPU computing capability demand $\mathrm{cpu}(n^v)$ and location demand $\mathrm{loc}(n^v)$. C_e^v is the attributes set of each virtual link consisting of available bandwidth capability demand $b(e^v)$.

2.3. Virtual Network Mapping Problem Modeling. The virtual network mapping problem is defined as mapping a virtual network G^v to $G^{s'}$, which is a subset of G^s. The CPU, location, and bandwidth constraints in G^v are all needed to be satisfied. The virtual network mapping can be naturally defined by $f_N : (N^v, E^v, C_n^v, C_e^v) \rightarrow (N^{s'}, P^{s'})$, where $N^{s'} \subset N^s$ and $P^{s'} \subset P^s$.

2.4. Virtual Network Survivability Analysis. In literature [22], Todimala and Ramamurthy have proved that the routing of virtual topology G^v over physical topology G^s is survivable if and only if $G^{v,i}$ is the node set that the virtual node corresponding to the substrate node with failure has been removed and the primary-cut set $\mathrm{PCS}_{G^{v,i}}\langle S, N^{v,i} \setminus S\rangle$ of the virtual topology exists at least one link not routed over any of the links passing through the substrate node with failure. $N^{v,i}$ is the virtual node set of $N^v - \{i\}$. $\langle S, N^{v,i} \setminus S\rangle$ is a cut set of G_v without node i, which divides the set $N^{v,i}$ into set S and set $N^{v,i} \setminus S$. The cut set $\mathrm{CS}_{G^{v,i}}\langle S, N^{v,i} \setminus S\rangle = \{(p, q) \mid (p, q) \in E \wedge (p \in S) \wedge (q \in N_{v,i} \setminus S)\}$. It also classifies cuts of the graph into primary-cuts and secondary-cuts based on the connectedness of the partitions. A cut is called a primary-cut if and only if both of the induced subgraphs are connected components. For the primary-cut $\mathrm{PCS}_{G^{v,i}}\langle S, N^{v,i} \setminus S\rangle$, the subgraphs that formed by node sets S and $N^{v,i} \setminus S$ are connected components. This conclusion provides inspiration for virtual network survivability in our reliable virtual network mapping method.

3. Integer Linear Programming Formulation

In this paper, we formulate the reliable heuristic mapping problem as an integer linear program in the following manner with the objective function of minimizing the cost of the substrate network.

3.1. Variable Description. (1) η_i^m, a binary variable, denotes the mapping relationship between virtual nodes and substrate nodes. If the virtual node m is mapped to the substrate node i, $\eta_i^m = 1$, and $\eta_i^m = 0$ otherwise.

(2) φ_{ij}^{mn}, a binary variable, denotes the mapping relationship between virtual links and substrate links. If the virtual node m is mapped to the substrate node i, $\varphi_{ij}^{mn} = 1$, and $\varphi_{ij}^{mn} = 0$ otherwise.

3.2. Objective Function. Consider

$$\min \sum_{(i,j)\in A_e^s} \sum_{(m,n)\in C_e^v} \varphi_{ij}^{mn} b\left(e_{mn}\right). \tag{1}$$

3.3. Constraints

3.3.1. Node Constraints. Consider

$$\forall i \in N^s,\ \forall m \in N^v, \quad \eta_i^m \operatorname{cpu}(m) \leq \operatorname{cpu}(i), \tag{2}$$

$$\forall i \in N^s,\ \forall m \in N^v, \quad \eta_i^m \left\|\operatorname{loc}(m) - \operatorname{loc}(i)\right\| \leq D(m), \tag{3}$$

$$\forall i \in N^s, \quad \sum_{m\in N^v} \eta_i^m \leq 1, \tag{4}$$

$$\forall m \in N^v, \quad \sum_{i\in N^s} \eta_i^m = 1. \tag{5}$$

3.3.2. Link Constraints. Consider

$$\forall (i,j) \in N^s,\ \forall (m,n) \in N^v, \quad \varphi_{ij}^{mn} b\left(e_{mn}\right) \leq b\left(e_{ij}\right), \tag{6}$$

$$\forall i \in N^s, \quad \forall (m,n) \in E^v,$$
$$\sum_{(i,j)\in E^s} \varphi_{ij}^{mn} - \sum_{(j,i)\in E^s} \varphi_{ji}^{mn} = \begin{cases} 1 & \text{if } \eta_i^m = 1 \\ -1 & \text{if } \eta_i^n = 1 \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

3.3.3. Survivability Constraints. Consider

$$\forall i \in N^s, \quad \forall N_{v,i} \in N^v,$$
$$\sum_{(m,n\in \mathrm{CS}(\mathrm{PCS}(S,G_{v,i}\setminus S)))} \varphi_{ij}^{mn} + \varphi_{ji}^{mn} < \left|\mathrm{PCS}\left(S, N_{v,i} \setminus S\right)\right|. \tag{8}$$

Remarks. (1) The objective function (1) is to minimize the cost of the link mapping. For a specific virtual network request, the resources allocated to the virtual nodes are identical. However, the resources assigned to the virtual links depend on the length of the substrate links. For this reason, we use the bandwidth usage of the substrate links to measure the cost of mapping.

(2) Constraint (2) means that the rest available CPU resources of substrate nodes should meet the demand constraints of virtual nodes. Constraint (3) means the locations of the substrate node loc(i) and the virtual node loc(m) should be at a distance within the range of $D(m)$. $\|\operatorname{loc}(m) - \operatorname{loc}(i)\|$ denotes the distance between m and i.

(3) Constraints (4) and (5) enforcing each virtual node in a virtual network request can only be assigned to one substrate node. Any two virtual nodes cannot be assigned to the same substrate node. Virtual nodes from different virtual networks can be assigned to the same substrate node.

(4) Constraint (6) indicates that the virtual link can be mapped to the substrate link which needs to satisfy the condition that the rest available bandwidth resource of the substrate link should meet the demand constraints of the virtual link.

(5) Constraint (7) represents link connectivity constraints. If virtual nodes m and n are mapped to the substrate nodes i and j, the virtual link (m, n) is mapped to the substrate link $p \in P^s(i, j)$ between i and j. If the node i is the source node, m is mapped to i and $\eta_i^m = 1$. The inflow of the source node is 0 and the outflow is 1. If the node i is the destination node, n is mapped to i and $\eta_i^n = 1$. The inflow of the destination node is 1 and the outflow is 0. For other nodes, the inflow and outflow are both 0. In brief, the link connectivity constraint equations above can be expressed as (7).

(6) Constraint (8) represents the survivability constraints of virtual networks. Let $|\mathrm{PCS}(S, N^{v,i} \setminus S)|$ denote the number of virtual links in the primary-cut set $\mathrm{PCS}(S, N^{v,i} \setminus S)$. For the virtual network with the removal of node i, the number of virtual links in primary-cut set that mapped to the same substrate link should be less than the number of virtual links in the primary-cut set, which means the virtual links in the same primary-cut set should not be mapped to the same substrate link. If the virtual links in the same primary-cut set are mapped to the same substrate link, substrate node failures will lead to the failures of links that pass through the substrate node. It means that if we remove all the links in the primary-cut set, the figure will no longer keep its connectivity.

4. A Hybrid Reliable Heuristic Mapping Method Based on Survivable Virtual Networks

For the integer linear programming of the reliable virtual networks mapping problem, we can quickly calculate the optimal solution when solution space is small. With the increase of the solution space, it is exponentially difficult to evaluate the optimal solution [25, 26]. In order to solve this problem, we propose a reliable heuristic virtual network mapping method called Hybrid-RHM-SVN using the ant colony algorithm proposed by Dorigo et al. [27], which has global optimization capability to calculate the approximate optimal solution. Hybrid-RHM-SVN consists of two steps: firstly, we use the enumeration strategy proposed by Tsukiyama, which has a linear time complexity, to calculate the cut sets of the virtual network that removes each node when a virtual network request arrives [28]. We find the primary-cut based on the approach proposed by literature [22]. Secondly, we use the ant colony optimization algorithm to achieve the approximate optimal mapping solution with the purpose of reducing the cost of the substrate network. In detail, when a virtual node is mapped to the substrate network, for the subgraph with the removal of the node, the edge in the primary-cut set should select the substrate path that does not pass through the substrate node corresponding to the virtual node that has been removed. When the node failure occurs, we can ensure the virtual network survivability and minimize the links failure caused by the node failure.

4.1. Nodes and Links Allocation Strategy

4.1.1. Nodes Allocation Strategy

Step 1. According to the location constraint of the virtual node m, we get the set $\theta(m)$, in which substrate nodes can be assigned to m, and $\theta(m) = \{i \in N^s \mid \|(\mathrm{loc}(m) - \mathrm{loc}(i))\| \leqslant D(m)\}$. $\theta(m)$ is a substrate node set in the circle with $\mathrm{loc}(m)$ as the center and $D(m)$ as the radius.

Step 2. There is no substrate node that can meet the location constraint of m when $\Omega(m) = \theta(m) \cap \{i \in N^s \mid \eta_i^m = 1\}$ is an empty set, which means the virtual network request is rejected; otherwise turn to Step 3.

Step 3. Obtain the value of $H(i')$ for the substrate node i' that can meet the node constraints in $\Omega(m)$,

$$H\left(i'\right) = \omega_1 \mathrm{cpu}\left(i'\right) + \omega_2 \sum_{j\in\Omega(m)} b\left(P_{i'j}\right), \tag{9}$$

where $P_{i'j}$ is the path between i' and its adjacent node j. ω_1 and ω_2 are the weights of the available CPU and bandwidth.

Step 4. Allocate the substrate nodes in $\theta(m)$ for the virtual node with the value of $H(i')$ decreasing.

4.1.2. Links Allocation Strategy

Step 1. If the virtual node m is assigned to the substrate node i, calculate the primary-cut set $\mathrm{PCS}_{G^{v,m}}\langle S, N^{v,m}\backslash S\rangle$ of the virtual network with removal of m.

Step 2. For the virtual link (p, q) in the primary-cut set, suppose that p and q are assigned to the substrate nodes p' and q', respectively. Find the substrate link in the loop-free path set $P^s(p', q')$ for (p, q). If there is a link $p \in P^s(p', q')$ not passing through the node i and satisfies the link constraints (6)~(8), we assign links for (p, q). If the links in the primary-cut set do not satisfy not passing through the node i, the virtual network does not have the survivability. We assign the link p' for (p, q) and p' should satisfy the link constraints (6) and (7).

4.2. Nodes and Links Mapping Based on Ant Colony Optimization. The ant colony optimization algorithm is a random search optimization algorithm that simulates ants foraging by iterations [29]. In the ant colony optimization algorithm, ants can release pheromone information when looking for food on the path. The pheromone has a certain scope to cause the attention of other ants. If the food is in a short distance, the ants in the path may be back and forth many times and leave more pheromone. The more pheromone is left, the higher probability of this path can be chosen. We use the ant colony optimization algorithm to solve the mapping problem, and the fitness function, transition probability, and pheromone trail update need to be redefined as follows.

Definition 1 (fitness function). The fitness function of the ant colony optimization algorithm fit is related to the objective function, which is the cost of mapping:

$$\mathrm{fit} = \sum_{(m,n)\in A_e^s} \sum_{(i,j)\in C_e^v} \varphi_{ij}^{mn} b\left(e_{mn}\right). \tag{10}$$

Definition 2 (transition probability). The transition probability is the probability that the virtual node m can be mapped to the substrate node i. The ant's transition probability at time t is related to the pheromone trail:

$$p_{mi} = \frac{\tau_{mi}(t)}{\sum_{x\in N^s} \tau_{mx}(t)}. \tag{11}$$

Definition 3 (pheromone trail update). In each iteration, the pheromone trail of virtual node m is mapped to the substrate node i updates as follows:

$$\begin{gathered} \tau_{mi}(t+1) = \left(1-\rho\right)\tau_{mi}(t) + \Delta\tau_{mi}^{\mathrm{best}}, \\ \Delta\tau_{mi} = \frac{Q}{f\left(s^{\mathrm{best}}\right)}, \end{gathered} \tag{12}$$

where ρ is the evaporation of trail, $0 < \rho < 1$. s^{best} is the optimal solution of the current iteration, and Q is a positive constant.

During the execution of mapping, the rest of available CPU resources and location of the substrate nodes should meet the demand constraints of the virtual nodes in (2)~(5). The virtual links can be mapped to the substrate links needing to satisfy the condition that the rest available bandwidth resources of the substrate links should meet the demand constraints of virtual links in (6) and (7). The details of the Hybrid-RHM-SVN algorithm are described as follows.

Step 1. Initialize the population size of ant SN and the total number of iterations N.

Step 2. Update the transition probability according to the pheromone trail, maps virtual nodes and links in accordance with nodes and links allocation strategy in Section 4.2, respectively.

Step 3. Obtain the fitness value fit for each artificial ant. If the fitness value in the current iteration is lower, the new mapping solution is better than the original. Use the new mapping solution instead of the original optimal solution s^{best}.

Step 4. Get the current optimal solution and update pheromone trail. If the current iteration is less than N, turn to Step 2; otherwise end the Hybrid-RHM-SVN algorithm.

4.3. Failed Nodes and Links Recovery Strategy. If the failure of the substrate node i occurs, all the virtual nodes m_i, $i \in \{1, 2, \ldots, N\}$, mapped on node i are affected. For m_i in the virtual network with survivability, the rest of the mapped virtual networks remain connected. We just need to repair the node m_i and links connected to this node. We reselect the substrate node for node m_i according to Step 4 of nodes allocation strategy and substrate links for virtual links connected to the node m_i according to the Step 2 of links allocation strategy. If the nodes and links do not satisfy the constraints of the failed virtual network nodes and links, the recovery fails and moves the virtual network to the end of the queue waiting for the next round of repairing. If the virtual networks do not have survivability, we use topology-awareness and reoptimization mechanism for virtual network embedding in literature [20] to remap the nodes and links. In links allocation strategy, if the node and links do not satisfy the constraints of the failed virtual network node and links, the virtual network is also moved to the end of the queue waiting for the next round of repairing.

5. Performance Evaluation

5.1. Simulation Environment. As specified in [7–21], we use the following standard setup for all our simulations. The graphs for the substrate network and virtual network requests are generated by using the GT-ITM topology generator tool [30], which is widely used for generating network topologies. The substrate network has 200 nodes and each pair of the nodes is randomly connected with a probability of 0.5, corresponding to a medium sized Internet service provider. The available CPU and bandwidth resources of the substrate nodes are real numbers uniformly distributed between 50 and 100. Virtual network requests arrive in a Poisson process with an average rate of 4 virtual network requests per 100 time units, and each request has an exponentially distributed lifetime with an average of 1000 time units. The number of the nodes in each virtual network request is determined by a uniform distribution between 4 and 10. Each pair of virtual nodes is also randomly connected with a probability of 0.5. The bandwidth requirement of the virtual link is a real number uniformly distributed between 0 and 40. A single node failure arrives in a Poisson process with an average rate of 2 per 100 time units. The horizontal ordinate and vertical ordinate of the node location are uniformly distributed between 0 and 100. Radius D is uniformly distributed between 50 and 100. Both the CPU resources weight α and bandwidth resources weight β are 1. The value of ω_1 and ω_2 is 1. The population size of artificial ant SN is 50 and the total number of iterations N is 100. Pheromone trail τ_{mi} is 5. ρ and Q are 0.8 and 10000, respectively.

Our simulation experiments evaluate three algorithms listed in Table 1. SVNE-ORP is based on the protection mechanism and SVNE-TAP is based on the recovery mechanism.

Table 1: Comparisons.

Notation	Algorithm description
Hybrid-RHM-SVN	Hybrid reliable heuristic mapping method based on survivable virtual networks
SVNE-ORP [19]	Reliable mapping method based on optimization sharing redundancy mechanism
SVNE-TAP [20]	Reliable mapping method based on topology-awareness and reoptimization mechanism

5.2. Evaluation Metrics. We define the reliable virtual network mapping evaluation metrics as follows.

5.2.1. Acceptance Rate of Virtual Network Requests. The acceptance ratio measures the percentage of virtual network requests accepted by the mapping algorithm over a period of time. Similar to the previous works, the acceptance rate of virtual network requests η_{acc} can be formulated by

$$\eta_{\text{acc}} = \lim_{T \to \infty} \frac{\sum_{t=0}^{T} \text{VNR}_{\text{acc}}}{\sum_{t=0}^{T} \text{VNR} + \varepsilon_1}, \tag{13}$$

where VNR_{acc} is the virtual network requests that has been mapped successfully. VNR is the virtual network requests that arrived. ε_1 is a positive number close to zero infinitely.

5.2.2. Average Revenue of Mapping Virtual Networks. The revenue gives us a clear insight into how much infrastructure providers will gain when accepting a single virtual network request. The average revenue of mapping virtual networks η_{rev} denotes how much infrastructure providers will gain in the long run:

$$\begin{gathered} R\left(G^{v}, i\right) = \alpha \sum_{n^{v} \in N^{v}} \text{cpu}\left(n^{v}\right) + \beta \sum_{e^{v} \in E^{v}} b\left(e^{v}\right), \\ \eta_{\text{rev}} = \lim_{T \to \infty} \frac{\sum_{t=0}^{T} R\left(G^{v}, i\right)}{T + \varepsilon_2}, \end{gathered} \tag{14}$$

where $R(G^{v}, i)$ is the revenue of mapping one virtual network successfully. α and β are weights of CPU and bandwidth capability. ε_2 is a positive number close to zero infinitely.

5.2.3. Recovery Rate of Virtual Networks. The recovery rate of virtual networks, which is the ratio of the number of virtual networks that has been successfully recovered to the number of virtual networks that is affected by the substrate failure, is an important evaluation of reliable virtual network mapping. The recovery rate of virtual networks η_{suc} can be formulated by

$$\eta_{\text{suc}} = \lim_{T \to \infty} \frac{\sum_{t=0}^{F(T)} \kappa'\left(t_i\right)}{\sum_{t=0}^{F(T)} \kappa\left(t_i\right) + \varepsilon_3}, \tag{15}$$

where $F(T)$ is the failure occuring in the substrate network between 0 and T. $\kappa(t_i)$ is the number of virtual networks that

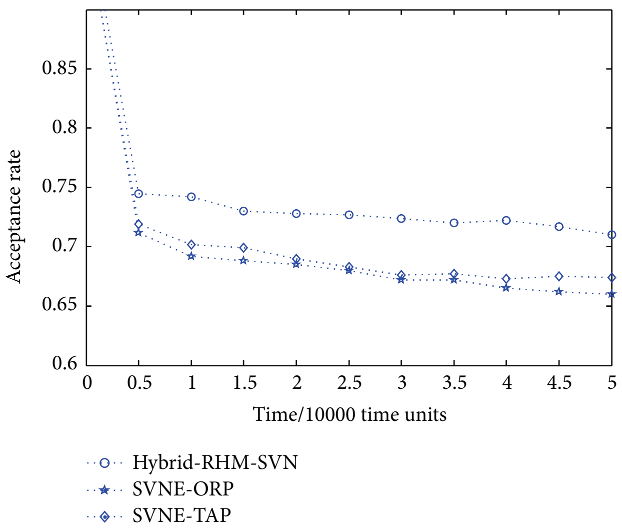


Figure 1: Acceptance rate of virtual network requests.

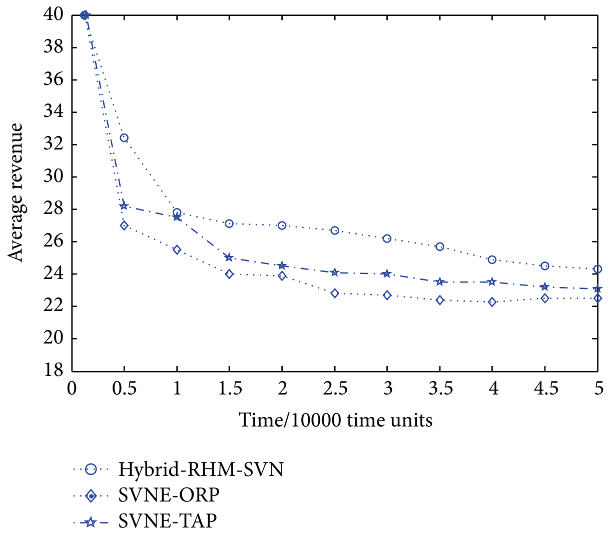


Figure 2: Average revenue of mapping virtual networks.

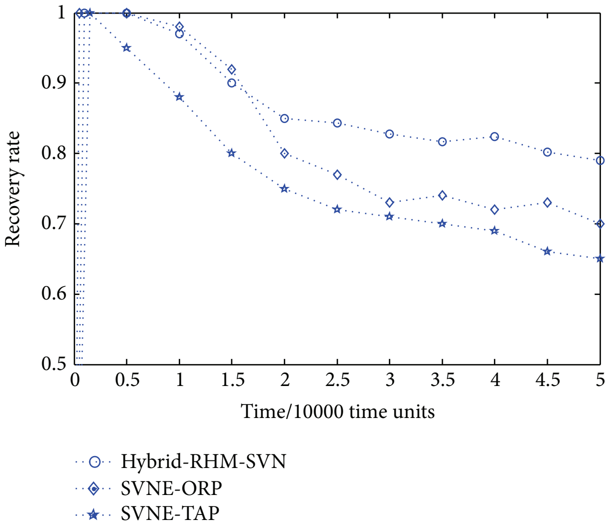


Figure 3: Recovery rate of virtual networks.

is affected by the substrate failure at $t_i \in (0, 1, \ldots, F(T))$. $\kappa'(t_i)$ is the number of virtual networks that has been recovered successfully. ε_3 is a positive number close to zero infinitely.

5.3. Evaluation Results. Figure 1 depicts the comparisons of virtual network request acceptance rates between our Hybrid-RHM-SVN algorithm and the existing algorithms. Due to the searching optimization capability of the ant colony optimization algorithm, Hybrid-RHM-SVN achieves substantially higher virtual network requests acceptance rate compared with those of the other algorithms in the long run. The Hybrid-RHM-SVN's acceptance rate is steady at 0.72 which is higher than SVNE-TAP. SVNE-ORP needs to provide redundant resources for virtual nodes and links, which will make excessive consumption of resources in substrate network. It is difficult to deal with the continuous arrival of virtual network requests.

Figure 2 depicts the average revenue between our Hybrid-RHM-SVN and other algorithms. It shows that Hybrid-RHM-SVN can achieve the highest average revenue in a reasonable number of iterations. Due to the highest virtual network requests acceptance rate, Hybrid-RHM-SVN algorithm can successfully map the largest number of virtual networks per unit time and the average revenue will be increased and steady at 25. In order to ensure the recovery rate of virtual networks, SVNE-ORP needs to provide many redundant resources for virtual network resources, leading to the lowest average revenue of mapping.

Figure 3 shows the recovery rates of virtual networks that are affected by single substrate node failure. Our experiments show that SVNE-ORP obtains the highest recovery rate followed by Hybrid-RHM-SVN in the time units from 0 to 16500. Hybrid-RHM-SVN obtains the highest recovery rate in the rest time. In the time units from 0 to 16500, the number of virtual networks needing to be mapped is less. There are enough available resources for SVNE-ORP to provide redundant resources for virtual nodes and links. Therefore, SVNE-ORP algorithm has a higher recovery rate in the early period. For Hybrid-RHM-SVN, not all the virtual networks have survivability. Virtual networks with survivability can be quickly recovered. Virtual networks without survivability need to be remapped by node migration and links remapping. In node migration and links remapping stage, the capacity constraints of nodes and links can be easily satisfied. But the location constraint loc(i) limits the number of substrate nodes that can be selected, which make the recovery rate of Hybrid-RHM-SVN lower than SVNE-ORP. With the substrate network resources occupied continually, there are fewer and fewer redundant resources that can be provided to virtual networks. The recovery rate of the SVNE-ORP algorithm will decrease rapidly. Although with the decrease of available resources in the substrate network, the recovery rate of Hybrid-RHM-SVN reduces slower than other algorithms

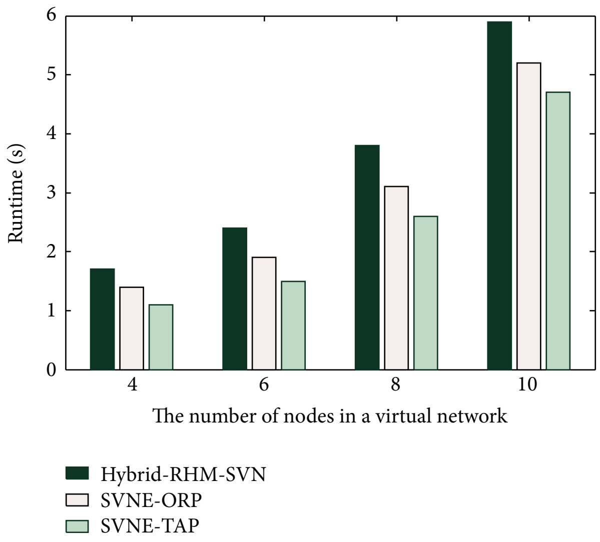


Figure 4: Runtime of mapping single virtual network.

due to the virtual network with survivability. The recovery rate is kept steady at 0.80, better than the other two algorithms in the long run.

Figure 4 shows the runtime of mapping single virtual network of the three comparisons. As shown in Figure 4, with the increasing of virtual nodes, the runtime of mapping single virtual network increases. SVNE-TAP consumes the most running time while Hybrid-RHM-SVN consumes the least running time. In Hybrid-RHM-SVN, calculating the primary-cut set will take a certain amount of time. But Hybrid-RHM-SVN handles virtual network requests in second class like the other two algorithms. Hybrid-RHM-SVN can increase the acceptance rate of virtual networks, the average revenue of mapping, and recovery rate of virtual network, which can meet the demands of infrastructure providers to deal with the virtual network requests.

6. Conclusion

The reliable virtual network mapping is one of the major problems in network virtualization researches. An efficient reliable mapping algorithm plays an important role in accepting more virtual network requests and generating more revenues for infrastructure providers. Different from the traditional protection mechanisms and recovery mechanisms, we proposed a hybrid reliable heuristic mapping method based on the survivable virtual network concepts inspired by survivable routing in wavelength division multiplexing WDM optical network. We formulate the reliable mapping problem as an integer linear program and propose a virtual networks survivable strategy. We propose a node allocation strategy, link allocation strategy, failed node, and link recovery strategy and use the ant colony optimization algorithm to achieve the approximate optimal mapping solution. The simulation results show that the survivable virtual network is a better idea of dealing with reliable mapping problem and our algorithm outperforms the previous approaches in the acceptance rate of virtual networks, the average revenue of mapping, and the recovery rate. In the future work, we will focus on how to optimize nodes and links allocation strategies to reduce the time usage of mapping.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This paper is supported by the National Natural Science Foundation of China (nos. 61370212, 61402127), the Research Fund for the Doctoral Program of Higher Education of China (no. 20122304130002), the Fundamental Research Fund for the Central Universities (no. HEUCFZ1213, HEUCF100601), and the Natural Science Foundation of Heilongjiang Province (no. ZD201102).

References

[1] N. Feamster, L. Gao, and J. Rexford, “How to lease the internet in your spare time,” *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 1, pp. 61–64, 2007.

[2] N. M. M. K. Chowdhury and R. Boutaba, “A survey of network virtualization,” *Computer Networks*, vol. 54, no. 5, pp. 862–876, 2010.

[3] J. S. Turner and D. E. Taylor, “Diversifying the internet,” in *Proceedings of the IEEE Global Telecommunications Conference*, pp. 755–760, IEEE, December 2005.

[4] N. M. M. K. Chowdhury and R. Boutaba, “Network virtualization: state of the art and research challenges,” *IEEE Communications Magazine*, vol. 47, no. 7, pp. 20–26, 2009.

[5] A. Bavier, N. Feamster, M. Huang, L. Peterson, and J. Rexford, “In VINI veritas: realistic and controlled network experimentation,” in *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM ’06)*, pp. 3–14, Association for Computing Machinery, 2006.

[6] D. G. Andersen, “Theoretical approaches to node assignment,” 2002, http://www.cs.cmu.edu/~dga/papers/index.html.

[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] X. Cheng, S. Su, Z. B. Zhang et al., “Virtual network embedding through topology-aware node ranking,” *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 2, pp. 39–47, 2011.

[9] G. Sun, H. Yu, V. Anand, and L. Li, “A cost efficient framework and algorithm for embedding dynamic virtual network requests,” *Future Generation Computer Systems*, vol. 29, no. 5, pp. 1265–1277, 2013.

[10] I. Houidi, W. Louati, and D. Zeghlache, “A distributed virtual network mapping algorithm,” in *Proceedings of the IEEE International Conference on Communications (ICC ’08)*, pp. 5634–5640, Beijing, China, May 2008.

[11] J. Lischka and H. Karl, “A virtual network mapping algorithm based on subgraph isomorphism detection,” in *Proceedings of the 1st ACM SIGCOMM Workshop on Virtualized Infrastructure*

Systems and Architectures (VISA '09), pp. 81–88, ACM, New York, NY, USA, 2009.

[12] M. Yu, Y. Yi, J. Rexford et al., "Rethinking virtual network embedding: substrate support for path splitting and migration," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 17–29, 2008.

[13] I. Fajjari, N. A. Saadi, G. Pujolle, and H. Zimmermann, "VNE-AC: virtual network embedding algorithm based on ant colony metaheuristic," in *Proceedings of the IEEE International Conference on Communications (ICC '11)*, pp. 1–6, IEEE, Kyoto, Japan, June 2011.

[14] X. Cheng, Z.-B. Zhang, S. Su, and F.-C. Yang, "Virtual network embedding based on particle swarm optimization," *Acta Electronica Sinica*, vol. 39, no. 10, pp. 2240–2244, 2011.

[15] Q. Zhu, H.-Q. Wang, H.-W. Lv, and Z.-D. Wang, "VNE-AFS: virtual network embedding based on artificial fish swarm," *Journal on Communications*, vol. 33, no. 1, pp. 170–177, 2012.

[16] S. Herker, A. Khan, and X. An, "Survey on survivable virtual network embedding problem and solutions," in *Proceedings of the 9th International Conference on Networking and Services*, pp. 99–104, 2013.

[17] M. R. Rahman and R. Boutaba, "SVNE: survivable virtual network embedding algorithms for network virtualization," *IEEE Transactions on Network and Service Management*, vol. 10, no. 2, pp. 105–118, 2013.

[18] H. F. Yu, C. M. Qiao, V. Anand, X. Liu, H. Di, and G. Sun, "Survivable virtual infrastructure mapping in a federated computing and networking system under single regional failures," in *Proceedings of IEEE Global Telecommunications Conference*, pp. 1–6, Miami, Fla, USA, December 2010.

[19] W. L. Yeow, C. Westpha, and U. C. Kozatl, "Designing and embedding reliable virtual infrastructures," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 2, pp. 57–64, 2011.

[20] N. F. Butt, M. Chowdhury, and R. Boutaba, "Topology-awareness and reoptimization mechanism for virtual network embedding," in *Networking 2010: Proceedings of the 9th International IFIP TC 6 Networking Conference, Chennai, India, May 11-15*, Lecture Notes in Computer Science, pp. 27–39, Springer, Berlin, Germany, 2010.

[21] Z. Cai, F. Liu, N. Xiao, Q. Liu, and Z. Wang, "Virtual network embedding for evolving networks," in *Proceedings of the 53rd IEEE Global Communications Conference (GLOBECOM '10)*, pp. 1–5, Miami, Fla, USA, December 2010.

[22] A. Todimala and B. Ramamurthy, "A scalable approach for survivable virtual topology routing in optical WDM networks," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 6, pp. 63–69, 2007.

[23] D. Zhou and S. Subramaniam, "Survivability in optical networks," *IEEE Network*, vol. 14, no. 6, pp. 16–23, 2000.

[24] S. Ramamurthy, L. Sahasrabuddhe, and B. Mukherjee, "Survivable WDM mesh networks," *Journal of Lightwave Technology*, vol. 21, no. 4, pp. 870–883, 2003.

[25] S.-D. Qing, J.-X. Liao, X.-M. Zhu, J.-Y. Wang, and Q. Qi, "Virtual network embedding algorithms in the network virtualization environment," *Journal of Software*, vol. 23, no. 11, pp. 3045–3058, 2012.

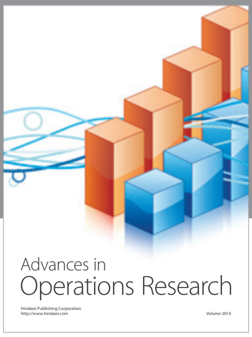
[26] A. Schrijver, *Theory of Linear and Integer Programming*, John Wiley & Sons, Chichester, UK, 1998.

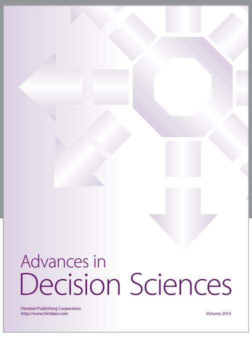
[27] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 26, no. 1, pp. 29–41, 1996.

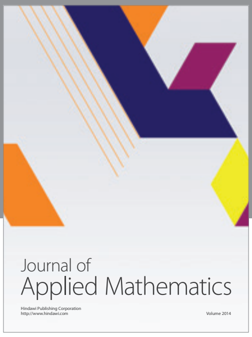
[28] S. Tsukiyama, I. Shirakawa, H. Ozaki, and H. Ariyoshi, "An algorithm to enumerate all cutsets of a graph in linear time per cutset," *Journal of the Association for Computing Machinery*, vol. 27, no. 4, pp. 619–632, 1980.

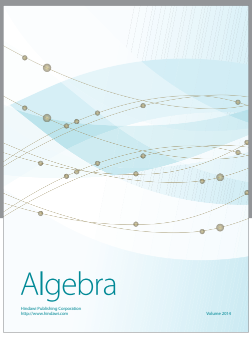
[29] M. Dorigo and C. Blum, "Ant colony optimization theory: a survey," *Theoretical Computer Science*, vol. 344, no. 2-3, pp. 243–278, 2005.

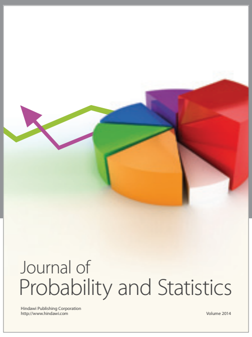
[30] Y. Xin, I. Baldine, A. Mandal, C. Heermann, J. Chase, and A. Yumerefendi, "Embedding virtual topologies in networked clouds," in *Proceedings of the 6th International Conference on Future Internet Technologies (CFI '11)*, pp. 26–29, ACM, June 2011.


Advances in
Operations Research

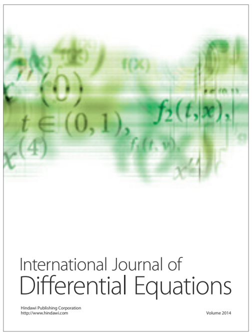

Advances in
Decision Sciences


Journal of
Applied Mathematics

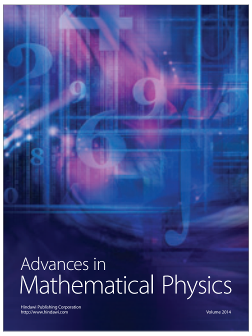

Algebra

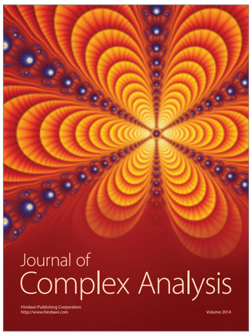

Journal of
Probability and Statistics

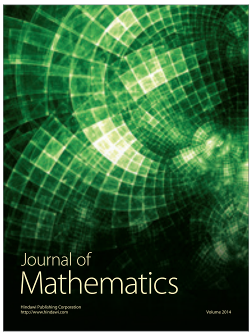
The Scientific
World Journal

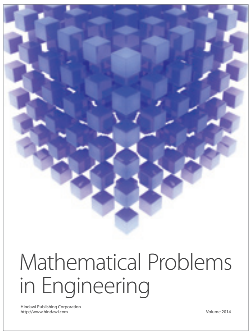

International Journal of
Differential Equations

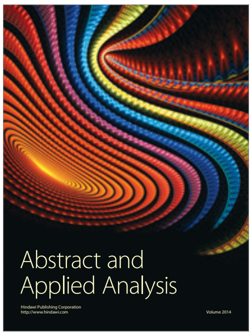
International Journal of
Combinatorics

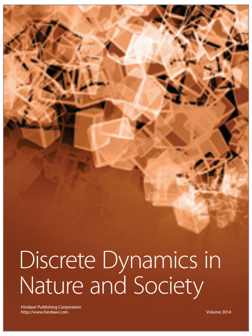

Advances in
Mathematical Physics

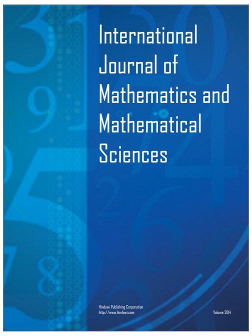

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

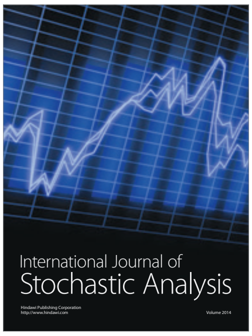

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization