LETTER

Survivable Virtual Network Topology Protection Method Based on Particle Swarm Optimization

Guangyuan LIU[†,††a)], ***Member and*** **Daokun CHEN**[†††], ***Nonmember***

SUMMARY Survivable virtual network embedding (SVNE) is one of major challenges of network virtualization. In order to improve the utilization rate of the substrate network (SN) resources with virtual network (VN) topology connectivity guarantee under link failure in SN, we first establishes an Integer Linear Programming (ILP) model for that under SN supports path splitting. Then we designs a novel survivable VN topology protection method based on particle swarm optimization (VNE-PSO), which redefines the parameters and related operations of particles with the embedding overhead as the fitness function. Simulation results show that the solution significantly improves the long-term average revenue of the SN, the acceptance rate of VN requests, and reduces the embedding time compared with the existing research results.

key words: *SVNE, VN topology connectivity, path splitting, ILP, particle swarm optimization*

1. Introduction

The emergence of network virtualization (NV) technology makes it easily to dispose new protocols and applications without affecting present networks, thereby effectively supporting network technology innovation [1]. One of main challenges in network virtualization is Virtual Network Embedding (VNE) [2], which finds the embedding of nodes or links in NV to physical infrastructure according to the resources and topology constrains. It has been proved to be NP-Hard problem [3].

Substrate link or node may fail. Because multiple links of VN can share one substrate path, one failure in substrate network may incur multiple VN links unavailable and increases obsession for the service providers (SPs). Thus providing reliability is significant to the virtual networks. The VNE can prevent failure in substrate network, which called the Survivable Virtual Network Embedding (SVNE) [4]. Most of SVNE research focus on strong survivable embedding guarantee with VN topology connected and resource recovery under substrate link or node failure [5]. They assume that SN-providers ensure survivability through over-provisioning the redundant resource for virtual links or nodes, which may result into extra overhead for the VN-providers. However, recently research [6] considers the weaker survivable mapping problem with VN topology connectivity guarantee in presence of substrate link failure. The advantage is that different quantity of failed VN link bandwidth can be provided according to customs' needs, and not simply providing fixed recovery bandwidth for VN links as traditional approaches. However, it still has the following problems: in the case that SN supports path splitting, the correct model is not given. The proposed embedding algorithm not only has large time overhead, but also the solution quality still has a large improvement space.

In this letter, we investigate how to improve the utilization rate of the SN resources with VN topology connectivity guarantee under single link failure in SN. The research in [3] has proved that the substrate network supports path splitting will have better resource utilization and may accept more VN request. Thus we first address an optimized Integer Linear Programming (ILP) model for VN topology connectivity guarantee problem in the case that substrate network supports path splitting. Then a novel survivable VNE solution based on Particle Swarm Optimization (VNE-PSO) is proposed. The solution encodes the virtual network embedding scheme into the location of the particles in the particle swarm, and uses the embedding overhead of the VN as the fitness function. The simulation results show that compared with the proposed solution in [6], the VNE-PSO significantly improves the long-term average revenue and VN request acceptance rate of SN. Meanwhile, it reduces the time overhead to obtain SVNE solution.

2. Problem Statement

2.1 The Substrate Network (SN) and Virtual Network Request (VNR)

The SN topology can be labeled as a weighted undirected graph $G_s = (N_s, L_s, C_s^n, C_s^l)$ where N_s denotes the set of substrate network nodes and L_s denotes the set of substrate network paths. C_s^n and C_s^l represent the set of attributes of substrate node n_s ($n_s \in N_s$) and link l_s ($l_s \in L_s$). Figure 1 (b) presents one example of SN.

Similar to the SN, the VNR can also be labeled as a weighted undirected graph $G_v = (N_v, L_v, R_v^n, R_v^l)$. N_v denotes the set of VN nodes and l_v denotes the set of VN links. R_v^n denotes the set of resource constraints of the virtual node n_v ($n_v \in N_v$), which is composed of the VN node capability

Manuscript received May 31, 2019.
Manuscript revised January 23, 2020.
Manuscript publicized March 4, 2020.

[†]The author is with School of Information Science and Technology, Shijiazhuang Tiedao University, Shijiazhuang, Hebei 050043, China.
[††]The author is with Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China.
[†††]The author is with Institute of Software Chinese Academy of Sciences, Beijing 100190, China.
a) E-mail: gyuanliu@163.com

DOI: 10.1587/transinf.2019EDL8113

Copyright © 2020 The Institute of Electronics, Information and Communication Engineers

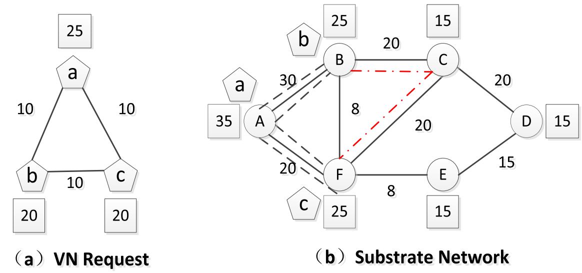

Fig. 1 Example of VN embedding

request and the location constrain $Loc(n_v)$; R_v^l denotes the virtual link l_v ($l_v \in L_v$) bandwidth resource request $BW(l_v)$. Figure 1 (a) presents one example of VN.

2.2 Virtual Network Embedding with VN Topology Connectivity Guarantee under Substrate Link Failure

VNE is defined by a embedding $M: G_v(N_v, L_v) \rightarrow G_s(N_s', P_s')$ from G_v to a subset of G_s, $N_s' \subset N_s$ and $P_s' \subset P_s$. Figure 1 (b) gives a virtual network embedding scheme for the virtual network request in Fig. 1 (a). The node embedding solution is $\{a \rightarrow A,\ b \rightarrow B,\ c \rightarrow F\}$. The link embedding solution is $\{(a,b) \rightarrow (A,B),\ (a,c) \rightarrow (A,F),\ (b,c) \rightarrow (B,A,F)\}$.

Since the bandwidth resource of each substrate path can be shared by many virtual links, a single substrate path failure may result into multiple virtual links failure. The topology of embedded VN may disconnect. For example, the failure of the substrate link (A,B) or (A,F) in Fig. 1 (b) can result into the topology of embedded VN disconnected in Fig. 1 (a). Therefore, the original embedding does not satisfy the survivable constrains. In contrast, if the virtual link (b,c) is mapped to (B,C,F), the VN topology remains connected in the presence of any single substrate link failure. We call such solution is virtual network embedding solution with VN topology connectivity guarantee under single link failure in SN. Note that the necessary condition for solution exists is that VN topology is redundant with 2-connected at least.

3. Integer Linear Programming Model with Supposing Path Splitting

In this section, we establish an Integer Linear Programming model for VN embedding with VN topology connectivity guarantee under single link failure in SN. Moreover, the SN supports path splitting in the model which can obtain better resource utilization and may accept more VN request.

The model involves the following two variables:

Binary variable f_{ij}^{uv}: In the embedding process, if the physical link (i,j) carries the virtual link (u,v), the variable takes a value of 1, otherwise the value is 0.

Binary variable x_i^u: In the embedding process, if the virtual node u is mapped to the physical node i, the variable takes a value of 1, otherwise the value is 0.

For a virtual network embedding request, the CPU overhead of different embedding schemes is the same, and the bandwidth overhead is different, so we use the following formula as the objective function of the model:

$$\text{Minimize} \sum_{(u,v)\in L_v} \sum_{(i,j)\in L_s} f_{ij}^{uv} \times BW(l_{uv}) \tag{1}$$

In terms of node constraints (2), the CPU capacity of the substrate network node must be able to meet the CPU capacity requirements of the virtual network node, and the substrate network node must be located within the D range from the location requested by the virtual node.

$$\forall u \in N_v,\ \forall i \in N_s,\ \begin{cases} x_i^u \times CPU(u) \le CPU(i) \\ x_i^u \times Dis(Loc(i), Loc(u)) \le D \end{cases} \tag{2}$$

In terms of link constraint (3), due to path splitting, the bandwidth of substrate paths mapped should satisfy the VN link bandwidth request. k^{e^v} denotes the number of splits for the virtual link $e^v \in L^v$. e^s denotes the any substrate path which the virtual link e^v will map. The formal description are as follows:

$$\forall e^s \in E^S: \sum_{e^v \in E^V} \frac{b(e^v)}{k^{e^v}} \times Y(p^{e^v}, e^s) \le b(e^s) \tag{3}$$

If virtual node u and v are mapped to the substrate network node i and j respectively, the virtual link (u,v) will be mapped to one of substrate paths from node i to node j during the virtual network embedding phase. At source i, the outgoing tragic is 1, the incoming traffic is 0. Therefore $\sum_{(i,j)\in L_s} f_{ij}^{uv} - \sum_{(j,i)\in L_s} f_{ji}^{uv} = 1$. At the junction j, the outflow is 0 and the inflow is 1. Therefore, $\sum_{(i,j)\in L_s} f_{ij}^{uv} - \sum_{(j,i)\in L_s} f_{ji}^{uv} = -1$. Flow conversation constraints are as follows:

$$\forall i \in N_s,\ \forall (u,v) \in L_v,$$
$$\sum_{(i,j)\in L_s} f_{ij}^{uv} - \sum_{(j,i)\in L_s} f_{ji}^{uv} = \begin{cases} 1, & \mathit{if} \quad x_i^u = 1 \\ -1, & \mathit{if} \quad x_i^v = 1 \\ 0, & \mathit{otherwise} \end{cases} \tag{4}$$

Previous work [7] has proved that a routing is survivable if and only if no single substrate link is shared by all logical links belonging to a cut-set of the logical topology. Correspondingly, in the context of VN embedding, if all the links in a cut-set of a VN are not sharing any substrate link, any single substrate link failure cannot lead to be not connectivity of the VN. VN topology connectivity constraint is as follows:

$$\forall (i,j) \in L_s,\ \forall M_v \in N_v,\quad \sum_{(u,v)\in CS(M_v - M_v)} f_{ij}^{uv} + f_{ji}^{uv} <$$
$$|PCS(M_v, N_v - M_v)| \tag{5}$$

Constraint (5) states that not all virtual links belonging to a cut-set can be carried on a single substrate link. In this constraint, only primary cut-sets $PCS(M_v, N_v - M_v)$ of the VN topology are considered instead of all cut-sets. This reduces the number of constraints and results in improved performance in terms of time [7]. $|PCS(M_v, N_v - M_v)|$ equals the number of edges in the primary cut-set, i.e., the size of this cut-set.

4. VNE-PSO Solution

4.1 PSO Foundation

The particle swarm optimization algorithm is a global random search algorithm based on swarm intelligence proposed by Kennedy and Eberhart. Compared with similar optimization algorithms, it has the advantages of fast execution speed and high efficiency. The particle speed and update formula are as follows:

$$V_{i+1} = wV_i + c_1 r_1 (X_{pb} - X_i) + c_2 r_2 (X_{gb} - X_i) \tag{6}$$

$$X_{i+1} = X_i + V_{i+1} \tag{7}$$

Where X_i represents the current position of the *i-th* particle, V_i represents its current velocity.

4.2 Redefinition of Particle Related Parameters and Operations

According to the optimized virtual network embedding problem model (in Sect. 3), the position, velocity, and related operations of the particles in the particle swarm are redefined as follows:

Definition 1 Location of the particle: The position vector of the particle $X_i = \left[x_i^1, x_i^2, \ldots, x_i^H\right]$ is defined as the *i-th* possible embedding scheme. H indicates the number of VN node in VN request. x_i^j takes a positive integer which value represents the substrate network node number selected by the *j-th* virtual node from list of substrate network candidate nodes.

Definition 2 The velocity of the particle: The velocity vector of the particle $V_i = \left[v_i^1, v_i^2, \ldots, v_i^N\right]$ is defined as the adjustment decision of the embedding scheme, which is used to guide the current embedding scheme to the better embedding scheme. v_i^j is a binary variable. If $v_i^j = 0$, it means that the *j-th* virtual node needs to reselect the node embedding from list of substrate network candidate nodes.

Definition 3 Subtraction Θ: $X_i \Theta X_j$ is used to calculate the difference between the two embedding schemes. If the embedding schemes X_i and X_j have the same value in the same dimension, the result of the difference is 0, otherwise 1.

Definition 4 Addition $\oplus$: $P_i V_i \oplus P_j V_j$ is used to obtain the adjustment decision of the embedding scheme. $P_i V_i$ and $P_j V_j$ represent the values of the dimensions of V_i maintained by the probability of P_i and the values of the dimensions of V_j by the probability of P_j respectively. $P_i + P_j = 1$ $(0 \le P \le 1)$.

Definition 5 Multiplication $\otimes$: $X_i \otimes V_i$ is used to obtain a new embedding scheme. The embedding scheme X_i adjusts its virtual node embedding scheme according to the adjustment decision V_i.

Therefore, we can obtain the basic formula for the position and velocity update of the redefined particle swarm optimization algorithm as follows:

$$V_{i+1} = P_1 V_i \oplus P_2 (X_{pb} \Theta X_i) \oplus P_3 (X_{gb} \Theta X_i) \tag{8}$$

$$X_{i+1} = X_i \otimes V_{i+1} \tag{9}$$

Where P_1, P_2 and P_3 are constants, and $P_1 + P_2 + P_3 = 1$.

4.3 VNE-PSO Solution

The VNE-PSO solution marks the fitness function (1) as $f(x)$, where the position vector X represents a possible embedding scheme. In the calculation of the fitness function, the node capability constraint is first checked according to Eq. (2), and Eqs. (3) (4) and (5) use the shortest path algorithm to check the feasibility of the current embedding scheme about bandwidth and connectivity constraints. In particular, If the embedding scheme is feasible, the value of $f(x)$ represents the overhead of the virtual network embedding. If the embedding scheme is not feasible, the value of $f(x)$ is set to $+\infty$;

5. Performance Evaluations and Analysis

The long-term average revenue and the VN request acceptance ratio were selected as evaluation indicators which were similar to previous work in [3]. The long-term average revenue is given by the following:

$$\lim_{\mathrm{T}\to\infty} \frac{\sum_{\mathrm{t}=0}^{\mathrm{T}} \mathrm{R}(\mathrm{G_v}, \mathrm{t})}{\mathrm{T}}.$$

where $\mathrm{R}(\mathrm{G_v}, \mathrm{t})$ is the revenue of accepting a VN request at time t can be defined by the following equation:

$$\mathrm{R}(\mathrm{G_v}, \mathrm{t}) = \sum\nolimits_{\mathrm{n_v}\in \mathrm{N_v}} \mathrm{CPU}(\mathrm{n_v}) + \sum\nolimits_{\mathrm{l_v}\in \mathrm{L_v}} \mathrm{BW}(\mathrm{l_v}),$$

where $\mathrm{CPU}(\mathrm{n_v})$ and $\mathrm{BW}(\mathrm{l_v})$ are the CPU and the bandwidth requirements for the virtual node $\mathrm{n_v}$ and link $\mathrm{l_v}$ respectively.

Moreover, the average execution time for embedding a VN request was also compared with the embedding algorithms S-CoViNE-ILP and S-CoViNE proposed in [6].

5.1 Experiment Setup

To quantify the efficiency of VNE-PSO for survivable VN request, we chosen the set of parameters conforms with the ones used in [3], [8]. The SN topology was configured to have 100 nodes and 500 links, a scale that corresponds to a medium-sized ISP. The node capacity and link bandwidth of substrate network were real numbers, which uniformly distributed of 50-100. We assumed that VN requests arrive following a Poisson process with an average arrival rate of 5 VNs per 100 seconds and each one had an exponentially distributed lifetime with an average of 500 seconds. For each VN request, the VN nodes were real numbers, which uniformly distributed of 2-20, and the degree of each VN nodes was set to 2-5 which ensures that the topology of the VN request is redundant. The node resource and link bandwidth

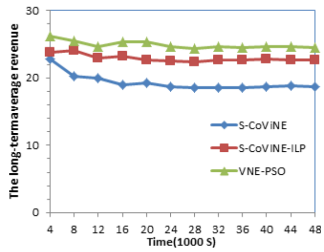

Fig. 2 The long-term average revenue

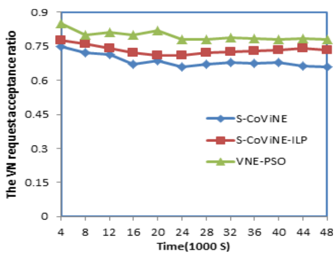

Fig. 3 The VN request acceptance ratio

resource requirements of VN were real numbers, which uniformly distributed of 0-50. The network topology and its location information were randomly generated by the GT-ITM tool. Each simulation run around 50,000 seconds with 2,500 VN requests. The SN supposed path splitting and any virtual link can be mapped to multiple substrate paths. However, in the experiment, we measured the impact of the number of splits k^{e^v} on VN embedding cost. The results showed that the embedding cost increased with an increasing number of splits. When the number of paths increased beyond 5, this behavior became more prominent. For this reason, we restricted the value of k^{e^v} between 2 and 5 for the experiments,

For the VNE-PSO algorithm, we set the size of the particle group N to 5. Meanwhile, the maximum number of iterations MG was 20. The P_1, P_2 and P_3 in the Eq. (7) were set to 0.1, 0.2 and 0.7.

5.2 Simulation Experiment Results and Analysis

Figure 2 and Fig. 3 showed that compared to S-CoViNE algorithm, VNE-PSO significantly improved the long-term average revenue (about 20%) and VN request acceptance rate (about 10%). Even compared to S-CoViNE-ILP algorithm, VNE-PSO still had an advantage obviously. For example, in the time 2000 seconds, the long-term average revenue increased around 8%. Meanwhile, the VN request acceptance rate increased around 7%. The main reason was that the solutions obtained by other two algorithms based on heuristic techniques were not optimal solutions. The VNE-PSO could obtain an approximate global optimal solution, which can significantly reduce the virtual network embedding overhead. Consequently, the possibility for the SN to accept more VNs was increased.

Figure 4 showed that VNE-PSO reduced the average

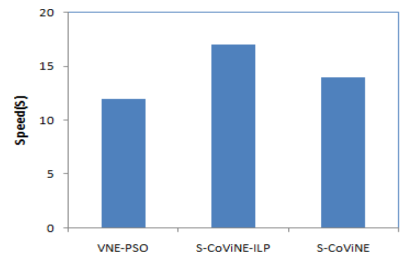

Fig. 4 Average execution time for a VN request

VN embedding time for a VN request around 15% compared to S-CoViNE, while reduced time around 30% compared to S-CoViNE-ILP. Because VNE-PSO was based on particle swarm optimization algorithm, which can effectively balance running time and solution quality by setting the number of particle swarms, the number of iterations and other iterative termination conditions.

6. Conclusion

With the SN supports path splitting, this letter established an Integer Linear Programming model for virtual network embedding with VN topology connectivity guarantee under substrate link failure. We also designed an effective solution based on particle swarm optimization. The solution redefined the position, velocity, and update operations of the particle based on the optimized mode. The simulation results showed that VNE-PSO significantly improved the long-term average revenue of the SN and the VN request acceptance rate while reducing the embedding time of virtual network compare with existing research results. However, as the network scale is enlarged, the VN embedding time will become longer until it can't meet the time requirements of the practical online embedding. For larger scale networks, we will design new algorithms to reduce the time cost in future work, such as using heuristic instead of cut-set calculating.

Acknowledgments

This work is support in part by National key research and development of China (No.2018YFB1701403), Natural science foundation of Hebei province (No.F2017210118), Colleges and universities in Hebei province science and technology research fund (No.QN2016270).

References

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

[2] A. Fischer, J.F. Botero, M.T. Beck, H. de Meer, and X. Hesselbach, "Virtual network embedding: A survey," IEEE Communications Surveys & Tutorials, vol.15, no.4, pp.1888–1906, 2013.

[3] M. Yu, Y. Yi, J. Rexford, and M. Chiang, "Rethinking Virtual Network Embedding: Substrate Support for Path Splitting and Migration," ACM SIGCOMM Computer Communication Review, vol.38, no.2, pp.17–29, 2008.

[4] M. Rahman, I. Aib, and R. Boutaba, "SVNE: Survivable virtual network embedding algorithms for network virtualization," IEEE Trans-

actions on Network and Service Management, vol.10, no.2, pp.105–118, 2013.

[5] S. Herker, A. Khan, X. An et al., "Survey on survivable virtual network embedding problem and solutions," Proc. ICNS, 2013.

[6] N. Shahriar, R. Ahmed, S.R. Chowdhury, M.M.A. Khan, R. Boutaba, J. Mitra, and F. Zeng, "Connectivity-aware virtual network embedding," IFIP Networking Conference (IFIP Networking) and Workshops, 2016, IEEE, pp.46–54, 2016.

[7] A. Todimala and B. Ramamurthy, "A scalable approach for survivable virtual topology routing in optical WDM networks," IEEE J. Sel. Areas Commun., vol.25, no.6, pp.63–69, 2007.

[8] X. Cheng, S. Su, Z. Zhang, H. Wang, F. Yang, Y. Luo, and J. Wang, "Virtual network embedding through topology-aware node ranking," ACM SIGCOMM Computer Communication Review, vol.41, no.2, pp.38–47, 2011.