PAPER

On the Optimal Approach of Survivable Virtual Network Embedding in Virtualized SDN

Rongzhen LI^{†a)}, Qingbo WU^{†b)}, Yusong TAN^{†c)}, *Nonmembers*, and Junyang ZHANG^{†d)}, *Member*

SUMMARY Software-defined networking (SDN) has emerged as a promising approach to enable network innovation, which can provide network virtualization through a hypervisor plane to share the same cloud datacenter network among multiple virtual networks. While, this attractive approach may bring some new problem that leads to more susceptible to the failure of network component because of the separated control and forwarding planes. The centralized control and virtual network sharing the same physical network are becoming fragile and prone to failure if the topology of virtual network and the control path is not properly designed. Thus, how to map virtual network into physical datacenter network in virtualized SDN while guaranteeing the survivability against the failure of physical component is extremely important and should fully consider more influence factors on the survivability of virtual network. In this paper, combining VN with SDN, a topology-aware survivable virtual network embedding approach is proposed to improve the survivability of virtual network by an enhanced virtual controller embedding strategy to optimize the placement selection of virtual network without using any backup resources. The strategy explicitly takes account of the network delay and the number of disjoint path between virtual controller and virtual switch to minimize the expected percentage of control path loss with survivable factor. Extensive experimental evaluations have been conducted and the results verify that the proposed technology has improved the survivability and network delay while keeping the other within reasonable bounds.

key words: cloud datacenter network, survivable virtual network embedding, network topology, virtual SDN, network delay

1. Introduction

With the rapid development of network technology and the requirement of providing network resources on-demand, virtual network embedding in virtualized SDN has attracted wide attention by the academia and industry. Virtual Network (VN) and Software Defined Network (SDN), which combine with each other and emerge as a prominent virtual network embedding approach, bring new strong motive power for cloud datacenter network. Virtualized infrastructure has been an important technology for a variety of heterogeneous types of applications in cloud datacenter, such as NFV, virtual CDN, delay-sensitive application and et al. VN can construct various types of different protocols and realize effective isolation of different networks. SDN has achieved the separation of data plane and control

plane and SDN hypervisor provides an essential virtualization for the logic separation from physical resources, such as FlowVisor [1] and OpenVertex [2], which can effectively control the network flow to divide the network resources into slices. Therefore, VN and SDN can be effectively combined to form a new method, which is beneficial to implement different network types and deliver NaaS [3] for satisfying the requirements of multi-tenants by sharing physical resources. Mapping VN onto cloud datacenter network usually includes virtual controller (vcontroller) mapping, virtual switch (vswitch) mapping and virtual link mapping, which is similar to traditional virtual network embedding and also NP-hard [4]–[7]. However, it is very low efficient to apply directly traditional virtual network embedding approach to solve VNE problem in virtualized SDN. Firstly, the novel VNE has its own virtual controller and the mapping problem should be solved preferentially. In addition, there are different important role of virtual nodes and links in VN. For example, the limitation of network delay of the data plane is not so rigid as to control plane.

Meanwhile, cloud datacenter no-doubt provides the ideal placement for the innovation of the virtualization technology. It is relatively more factors which need to be considered in virtualized SDN, such as vcontroller mapping and control path which is needed to consider virtual controller-to-switch link and its path diversity. The survivability of virtual network embedding is facing challenges because of the centralized control and the shard features of physical SDN. And virtual SDN controller mapping problem have a significant impact on the survivability of VN. In order to improve the performance and enhance the survivable virtual network embedding (SVNE), the virtualization of SDN should take fully advantage of network virtualization, which abstracts the logical network layer from physical network, and SDN that can validly control network flow.

Survivable virtual network embedding has drawn many researchers' attention. SVNE should guarantee that VN, which has been embedded into physical SDN network, works normally against physical component failure. Recently, the common strategies can be classified into two categories: protection mechanism and restoration mechanism. However, many research works adopt protection mechanism to protect the survivability of VNE but also take additional expenditure of physical resources. Some previous researches have focused on SVNE which is completely equivalent to nodes and links while few have considered the topology attributes of virtual network and the expected percent-

Manuscript received September 14, 2017.

Manuscript revised November 13, 2017.

Manuscript publicized December 18, 2107.

[†]The authors are with College of Computer, National University of Defense Technology, China.

a) E-mail: lirongzhen11@nudt.edu.cn

b) E-mail: qingbo.wu@nudt.edu.cn

c) E-mail: ystan@nudt.edu.cn

d) E-mail: zhangjunyang11@nudt.edu.cn

DOI: 10.1587/transinf.2017EDP7291

age of control path loss, and assume that the physical SDN is always operational. But, in fact, network node and link failures may occur at any time in the actual environment. The survivability of virtual network is neglected slightly, which may cause the vital malfunction and damage the profit of service providers and tenants. This paper studies the method of topology-aware survivable virtual network embedding, which deeply takes account of different importance of nodes (including vcontroller and vswitch) and links (including virtual controller-to-switch and virtual data link) and develops a SVNE approach, combined controller-to-switch multiple concurrent connection, path diversity and network delay to enable the VN embedding with the survival characteristics. The vcontroller and virtual controller-to-switch, for example, should be given priority to the performance while other trivial nodes/links can be slightly unsatisfied to reduce the difficulty of VN, which is beneficial to increase the number of virtual network requests.

In this paper, topology-aware survivable virtual network embedding is used to enhance the reliability of virtual network while keeping the embedding efficiency and revenue/cost (Rev/Cos) ratio of the system in an acceptable level. The major contributions in this paper are summarized as follows:

- 1 Introduce topology-aware survivable virtual network embedding approach by sharing cloud datacenter network, which enhance the survivability of virtual network.
- 2 Apply SVNE strategy with network delay and disjoint path between vcontroller and vswitch to minimize the expected percentage of control path loss through the shared physical SDN, which can significantly enhance the survivability.
- 3 The optimal controller location selection factor is used to improve the survivability of virtual network while little impacts acceptance ratio and revenue/cost ratio because network centrality is used to optimize the embedding.
- 4 Extensive experiments have been conducted and the proposed algorithm was compared with other similar algorithms

The rest of this paper is organized as follows. In Sect. 2, we discuss the related work. Section 3 describes SVNE Model and mathematical formulation. Section 4 develops the approach of the topology-aware survivable virtual network embedding. Section 5 provides survivable virtual network algorithms. In Sect. 6, we present the performance evaluation. Section 7 concludes this paper and looks forward to the future.

2. Related Work

Virtualization technology have drawn great attention in recent years, and many virtual network embedding (VNE) problems as well as solutions have also been studied extensively [8]–[10]. Meanwhile, the technology of VN is evolving with the development of cloud datacenter network and SDN. VNE is becoming a more complex problem and also an efficient solution that effectively allocates physical network resources to logical virtual network resources, which

can be reduced to constrained multi-objective optimization. Lots of researches have focused on designing and developing integral linear programming (ILP) and heuristics algorithm by simplifying the problem through various assumptions and many approaches of VNE have been proposed by researchers. However, innovative virtual network, which combines VN with SDN, are very different with traditional VN in the cloud datacenter, so that it is imperative for new methods to solve the problem of the novel virtual network embedding. On the other hand, VN becomes more fragile due to the same shared and the centralized physical SDN, the survivability of VNE still has not been solved well.

Several of the previous studies have proposed the approach that combines VN and SDN to solve virtual network embedding problems in cloud datacenter. E. Salvadori et al. [11] described an approach to embed virtual topologies based on OpenFlow by overcoming the problem that was strictly tied with the underlying physical topology, and links and nodes must match the substrate network configuration. Papagianni et al. [12] studied the virtual topology embedding approach for SDN-enabled networked cloud environments and adapted an existing virtual topology mapping algorithm to match the requirements of virtual SDN. LUO Shouxi et al. [13], based on Fat-Tree network topology, developed traffic-aware virtual data center embedding approach. Shui-qing GONG et al. [14] combined the controller placement with the virtual network embedding as a join vSDN embedding problem and formulated it into an ILP with multi-objectives, which is to minimize the embedding cost as well as the controller-to-switch delay for each vSDN. Gomes RL et al. [15] proposed a Bw-Risk-Ratio algorithm for VSDN allocation and used SDN together with VN to enhance the management, planning and resource usage of network.

However, most of current research about VNE do not enough consider to combine resilient network topology with control path to optimize the survivability of VN. In the traditional approach, one-stage embedding improves mapping quality but extends the embedding runtime. While two-stage embedding separates usually the node embedding and link embedding, many researchers do not take into account enough the relationship between nodes and links but network topology has a great influence on embedding efficiency of VN in cloud datacenter. Control path is the set of routes, which is used for communications between vswitches and vcontrollers and has an important influence on network delay and reliability. On the other hand, due to different resource constraints, SVNE should be researched for searching optimization solution. There are some existing algorithms that handle all the nodes or links equally and no difference in executing all the embedding, while the role of node or link may be different which should be clearly distinguished for virtual network embedding. Recently, some researchers append survivable features to VNE but do not explore deeply vcontroller mapping and just treat all node mapping equally. In other word, the proposed approach do not combine the network topology and control path to opti-

mize the VN survivability.

According to current research and existing problems, this paper develops a topology-aware SVNE method with survivable factor based on network centrality and control path, which combines network topology, disjoint path and network delay with the measuring strategy, and optimizes our previous proposed algorithm [16] to effectively improve VN survivability and enhances the tenants' experience.

3. SVNE Modeling and Problem Description

3.1 Virtual Network Embedding Modeling

Physical Cloud Datacenter Network The infrastructure provider owns physical cloud datacenter network, which is modelled as a weighted undirected graph $G_p = (N_p, L_p, An, Al)$, where N_p and L_p are, respectively, the set of physical nodes and links. Let C refer to the set of vcontroller placement in physical SDN and lc denote the total number of control paths which is selected by topology-aware shortest path method to reduce propagation delay (hops). While nodes and links are allocated with their constrained attributes, for example CPU, RAM, TCAM and bandwidth, denoted by An and Al . $An(c)$ refers to the CPU capability of physical node and $Al(b)$ represents the bandwidth of physical link.

Virtual Network Request Virtual Network Request (VNR) which is distinct from physical infrastructure is a logical topology of the tenants' requirements. VNR refers to the vertex and edge of the graph and also models as a weighted undirected graph $G_v = (V_v, E_v, R_v, Re)$, where V_v and E_v respectively represents with the set of virtual nodes and virtual links. R_v and Re are defined to the requirements of virtual node and virtual links. For instance, $R_v(c)$ refers to the CPU capability of virtual node and $Re(b)$ represents the bandwidth of virtual link.

Virtual Network Embedding As our previous work [16], VNE is usually defined to the constrained resources allocation that G_v embeds the subgraph of G_p that should satisfy the resource constraints of R_v and Re for VNR. VNE problem is a mapping process from G_v to G_p denoted as $M: G_v \rightarrow (N', P', R_n, R_l)$, N' contained in N_p , P' contained in P_p and P_p is set of L_p , R_n and R_l are the physical resources allocated to the virtual nodes and virtual links, respectively, VNE model is shown in Fig. 1. Figure 1 (a) represents Alice's VNR and Bob's VNR, where Alice's VN consists of one vcontroller and three vswitches (which is abbreviated V in Fig. 1), and Bob's VN consists of one vcontroller and two vswitches. Figure 1 (b) show that Two VNRs (Fig. 1 (a)) map into virtualized SDN, where the process is shown in Fig. 1 (c). Figure 1 (c), based on FlowVisor virtualization, describes the VNE on OpenFlow networks, where Alice and Bob's VNR adopt the same resource allocate policy to design and implement the virtual network embedding. In addition, this paper just assume all VNRs adopt the same policy of resource allocation for enhancing the virtual network survivability.

Fig. 1 Virtual network embedding model

Table 1 Notations of virtual network embedding problem

G_p	Physical Cloud Data Center Network
N_p	Nodes of Physical Cloud Data Center Network
L_p	Links of Physical Cloud Data Center Network
An	Node Attribute of Physical Cloud Data Center Network
Al	Link Attribute of Physical Cloud Data Center Network
P_p	Path on Physical Cloud Data Center Network
G_v	Virtual Network
V_v	Nodes of Virtual Network
E_v	Links of Virtual Network
R_v	Node Requirements of Physical Cloud Data Center Network
Re	Link Requirements of Physical Cloud Data Center Network
M	Function of Embedding the graph of VNR to the Physical graph
Fn	Function of Mapping Virtual Node to Physical Node
Fl	Function of Mapping Virtual Link to Physical Path

VNE can be divided into two phases:

Node mapping, which includes the mapping of vcontroller and vswitch, is defined as: $fn: V_v \rightarrow N_p$, mapping the virtual nodes to physical nodes where the arbitrary v_v belongs to V_v and existing n_p belongs to N_p , $n_p = fn(v_v)$, $An(c) \geq R_v(c)$ and $An(b) \geq R_v(b)$, the concept is shown in Table 1.

Link mapping, which includes the mapping of virtual link and controller-to-switch link, is defined as: $fl: E_v \rightarrow P_p$, mapping virtual links to physical path where the arbitrary e_v belongs to E_v and existing p_p belongs to the P_p , $p_p = fl(e_v)$ and $An(b) \geq Re(b)$, where the P_p consists of a subset of the physical path set L_p , p_p consists of a subset of the physical path L_p .

3.2 Objectives

Survivable Virtual Network Embedding is defined to be the optimization problem of the constrained multi-objective, which should embed virtual network requests as many as possible into the physical network while guaranteeing the network survivability and SLA. The objective of this paper is to maximize the reliability of the control network while keeping success rate and revenue/cost ratio within reasonable bounds.

In order to analyze the reliability of physical cloud datacenter network, the expected percentage of control path loss is defined to the impact of virtual SDN because of physical network component failure. We define p_g to be the

failure probability of network component g (the elements of graph) for each physical network component $g \in Np \cup Lp$ and assume that the failure of g leads to disjoint the d_g control path which is connected between vcontroller and vswitch. So that the expected percentage of control path loss is defined as

$$\theta = \frac{1}{lc} \sum_{g \in Np \cup Lp} d_g p_g \quad (1)$$

lc is the variables of the total number of control paths that is related to the arrival of virtual network request. Disjoint path [27] means that path is no shared physical node.

Similar to the previous work [17], the definition of revenue to successfully accept a VN at time t can be formulated by:

$$R(Gv(t)) = \alpha \sum_{e_v \in Ev} BW(e_v) + \beta \sum_{v_v \in Vv} CPU(v_v) \quad (2)$$

where $BW(e_v)$ and $CPU(v_v)$, respectively, represents the bandwidth and CPU requirements of Ev and Vv . α, β represents the balance factor of bandwidth and CPU.

Similar to [18], the cost of a request Gv is defined as the sum of resources allocated to VN requests at time t .

$$C(Gv(t)) = \alpha \sum_{e_v \in Ev} BW(e_v) * length(e_v) + \beta \sum_{v_v \in Vv} CPU(v_v) \quad (3)$$

where $length(e_v)$ refers to the length of the loop-free path and its value is greater the more link bandwidth is consumed.

The revenue to cost ratio is used to measure the quality of the virtual network embedding, which is defined as:

$$Rev/Cos = R(Gv(t))/C(Gv(t)) \quad (4)$$

According to (2) and (3), where the larger value means to be better embedding quality, the range of Rev/Cos value is between 0 and 1.

The runtime of virtual network embedding represents with the total embedding time of virtual nodes and virtual links.

$$sumT = \sum_{e_v \in Ev} T(e_v) + \sum_{v_v \in Vv} T(v_v) \quad (5)$$

where $T(e_v)$ and $T(v_v)$ refer to the consumed time by virtual links and virtual nodes embedding respectively.

In order to improve controller-to-switch delay, $Lavg(c^*)$ refers to the average propagation latency for a placement of controllers c^* . $d(v_v, c^*)$ is the shortest path from source to destination nodes. The average controller-to-switch latency is defined as:

$$Lavg(c^*) = \frac{1}{n} \sum_{v_v \in Vv} mind(v_v, c^*) \quad (6)$$

where $n = |Vv|$ is the number of nodes v_v and c^* is the set of

controller placement.

Ncl is defined to the max connectivity of virtual SDN embedding (part of VNE), which refers to the average of disjoint paths between forwarding devices and their controller instance.

$$Ncl = \max_{c \in C, s \in S} \frac{\sum C_{c,s}}{n} \quad (7)$$

where C is the set of controller, S is the set of switch, n is the same as Eq. (6) and $C_{c,s}$ represents the number of disjoint paths between controller c and switch node s .

4. Topology-Aware Survivable Embedding Approach

In order to effectively map virtual network requests into the physical cloud datacenter network and enhance virtual network survivability, this section describes the topology-aware survivable method together with network centrality, survivable factor and multipath. The survivable factor is inversely proportional to network delay and proportional to the number of disjoint path between controller and forwarding devices and multipath approach is used to the left virtual link to form the weak survivability of virtual network without using any backup resources.

4.1 Topology-Aware Virtual Network Embedding

Network topology is a logical architecture for the network components connection and every topology has its own features. Network centrality is an important metrics for the features of the topology and plays an important role in the region of complex network analysis, such as the location of the network node measured to the influence of entire network topology. There are several effective strategies to measure network centrality of the network in graph theory and the field of network analysis. Degree, betweenness, closeness and eigenvector [19] are commonly used to measure the centrality of network topology. The different characteristics of the network are measured by different attributes of the network centrality and different analytical results are produced for network topology. Betweenness centrality is mainly used to measure the switching capacity and used to network congestion control for traffic flow. The eigenvector refers to centrality the centrality of the direct adjacent network. This paper, taking into account the topology attributes of virtual network, makes use of closeness and degree of network topology to optimize virtual network embedding problem, and thus effectively coordinates the embedding of the virtual nodes with virtual links for virtual network requests.

On the other hand, the attributes of network topology has a relative closely relationship between the topology and the performance of the network services, for example, network function virtualization (NFV) including more components will need more physical resources and consumes more network performance. However, the existing related works do not sufficiently take into consideration of network topology, although some of the existing work [19], [20] refer to

the attributes of network topology and justly use capacity and bandwidth to assess the importance of node or link. For taking full advantage of the topological properties and improving the acceptance rate and increasing revenue, we, based on the network topology, further make use of the impact of the links and nodes with other adjacent to estimate the importance of the node/link and apply multipath to reduce the network bandwidth fragmentation and optimize the utilization of network bandwidth in this paper.

Meanwhile, controller placement is a special problem in traditional SDN. So it is necessary to give full consideration to the communication delay between critical nodes and other nodes in the novel virtual network, for example, controller-to-switch, which is an important impact on the performance and reliability of the entire network system [24], [27]. Closeness centrality is defined to be a global conception of the nodes in the network topology which refers to the proximity of the node and all other nodes. The approach of closeness centrality takes full consideration of the distance from a node to all other nodes. For example, there is network hop to represent distance among nodes. In order to indicate node weight, CPU and Memory are used to weight network nodes and links are expressed by the bandwidth, while degree centrality only takes into account directly adjacent to the edge nodes and is to measure the local centrality and the bandwidth. Thus, this paper make use of CPU as node weights and chooses the degree and closeness to be the optimal strategy of network centrality for virtual network embedding. There is an important conception of topological space in mathematics. Closeness, the two direct adjacent points, represents that the two points are arbitrarily near to each other in any case and is a sophisticated measure of centrality in the region of network analysis and graph theory. The closeness of a node n_i refers to the reciprocal of the sum of geodesic distances to all other reachable nodes. The formulation of closeness can be denoted as our proposed work [18]:

$$Cc(n_i) = 1 / \sum_{j=1}^n d(n_i, n_j) \quad (8)$$

$d(n_i, n_j)$ refers to the distance of the shortest path between nodes n_i and n_j . Whatever n_i is close to the central location of the network topology and its closeness value will be relatively large, or else the closeness of n_i is in boundary position that has a small value.

The degree of a node is the number of immediate adjacent links to its neighbors that can be explained by the immediate possibility that a node interconnects closely with others, the formulation of degree is defined as:

$$D(n_i) = \deg(n_i) \quad (9)$$

The degree is defined to the total bandwidth of the same adjacent links for a node, which is known as strength in a weighted network topology. The strength of node is defined as follows:

$$S(n_i) = \sum_{l \in L(n_i)} BW(l) \quad (10)$$

$L(n_i)$ is the set of adjacent links of n_i .

Combining strength (10) with closeness (8) and degree (9), this paper, similar to Newton's law of universal gravitation, formulates the amount of resources for node n_i as:

$$R_{dc}(n_i) = CPU(n_i)D(n_i)[Cc(n_i)^2] \sum_{l \in L(n_i)} BW(l) \quad (11)$$

where $d(n_i, n_j)$ has an inverse-square effect on its importance in the network, which is included in $Cc(n_i)$.

4.2 Virtual Network Embedding with Survivable Factor

Survivable virtual network embedding can be divided two parts: virtual SDN controller mapping and the left virtual network mapping which adopts multipath method to improve the efficiency of VNE. This session mainly introduces the survivability of the vcontroller mapping.

Virtual controller mapping is similar to the controller placement problem, and its location and control path selection has an important impact on the whole network survivability and network latency. In order to minimize the controller-to-switch delay and obtain the optimal network survivability, this paper adopts the survivable factor of the vcontroller mapping to enhance the network reliability. In addition, the mapping of the vcontroller will affect the consequent vswitch and virtual link mapping, so that our approach also balances the trade-off between vcontroller mapping and consequent vswitch and virtual link mapping. In order to achieve this goal, firstly, this paper applies the survivable factor to maximize the network survivability. The survivability of the virtual SDN depends mainly on the location of the selected physical nodes for virtual controller and control path diversity. Thus, according to (6) and (7), the survivable factor of the node is defined as

$$SF = \frac{N_{cl}}{L_{avg}(c)} \quad (12)$$

At the same time, the capacity of physical nodes is closely related to the performance of virtual controller nodes, only if the physical node resources meet the requirements of the virtual controller requests and its adjacent nodes have physical resources as many as possible, the location selection of the vcontroller is optimal. According to the mapping requirements of the vcontroller, the capacity of the physical node should choose the optimization. Coordinated with survivable factor and the node location selection, this paper defines the optimal controller location selection factor (*OLSF*). The *OLSF* of node n_i refers to

$$OLSF(n_i) = SF \cdot \sum_{n_j \in N_s} (CPU(n_j) + TCAM(n_j))bw(n_i, n_j) \quad (13)$$

where $bw(n_i, n_j)$ refers to the available bandwidth along with the path from n_i to n_j , TCAM, similar to Memory, reflects the capacity of switch.

The embedding of the vcontroller is directly related

to the location selection of other virtual nodes that are attached to the virtual controller, which has an impact on the whole virtual topology embedding of virtual network. Therefore, *OLSF* not only needs to take into account the controller-to-switch network delay but also the reliability of the controller-to-switch link. Meanwhile, *OLSF* also needs to balance the available resources of the physical nodes and the physical network bandwidth, which is conducive to the consequent virtual node and link mapping, thereby improving the acceptance rate of virtual network requests. The vcontroller mapping can be seen as a special virtual node embedding problem in virtual network embedding. So it is necessary to give full consideration of the communication delay between critical nodes and other nodes in virtual SDN. For example, the survivability of controller-to-switch link has an important impact on the performance and reliability of the entire network system.

5. Survivable Virtual Network Embedding Algorithm

This section describes the topology-aware survivable virtual network embedding algorithm—TSVNE, which is a two-stage virtual network embedding approach. In the first phase, TSVNE maps the virtual SDN controller and vswitch of VNR to the physical nodes. According to the mapping status of virtual nodes, and then it will decide whether corresponding virtual link is mapped into the physical link.

In order to optimize network delay and survivability of virtual SDN in the two-stage of VNE, we explore deeply the strategy of vcontroller mapping, which is similar to controller placement and has a few differences. Firstly, the virtual controller mapping is logically isolated and dynamically changing to come in or leave at any time, which is different from the controller placement problem of SDN in this perspective. Thus, it is difficult to select a number of candidate controller node location at a time and needed fully to adjust the metrics of *OLSF* to achieve the optimization of virtual SDN controller embedding. Meanwhile, in order to quantify virtual SDN survivability, the disjoint path and the average network delay of controller-to-switch are selected to be the survivable parameters and this strategy fully takes consideration of the relationship between adjacent nodes, so as to optimize the virtual SDN survivability and the whole embedding efficiency. The process is shown in algorithm 1.

Algorithm 1 shows the process of the vcontroller mapping of virtual network requests. Firstly, according to the formulation (13), the nodes of physical SDN are sorted by their *OLSF* weight in no-increasing order (lines 1–2). Lines 3–8 describes the virtual controller mapping into the physical nodes. When virtual network requests arrive, virtual nodes of VNRs are sorted in descending order, which is similar to physical network nodes, and then it will try to take the virtual controller with the largest weight to map the largest weight nodes of physical SDN network (line 4). If the CPU, TCAM and bandwidth of physical network satisfy the virtual SDN controller nodes, the virtual nodes are mapped successfully (lines 5–7). At last, attempting repeat-

edly to map the virtual controllers in the time window, the virtual SDN is mapped successfully only if all virtual controller nodes are mapped successfully (line 9). Otherwise, this VNR will be discarded.

The embedding method of virtual switch also has an important influence on the whole embedding of virtual network. The network topology can provide measurement parameters for the high-quality embedding. The virtual switch embedding fully considers the topology of virtual network based on degree and closeness. TSVNE combines the weight of the nodes to implement virtual switch embedding.

It is shown that the progressing of vswitch embedding is illustrated step by step in Algorithm 2. Firstly, DegreeandClosenessRank() is defined to the weight function of the nodes according to the formulation (11). The nodes of physical network are sorted by their weights in descending order and the vswitch is tried to embedded into physical nodes in the queue (lines 1–2). When virtual network requests arrive, the requested vswitches are sorted in descending order, which is similar to physical network nodes (lines 3–4). And then, the largest weight of virtual switch of virtual network request will be mapped to the largest weight nodes of physical network (lines 5–14). All the virtual nodes, which are included in VNR, will try to be mapped. If the virtual network request, which usually includes the CPU, TACM and bandwidth, is satisfied by unmapped physical resources, the vswitch are mapped successfully (lines 7–10). While the

Algorithm 1: Vcontroller Embedding Algorithm with OLSF

```

1: Calculate  $OLSF(n_i)$  of all residual physical nodes using (13) in no-increasing order
2: Rank physical nodes according to the calculating results
3: for all the left virtual nodes do
4:   Mapping virtual controller of  $G_{vi}$  to the physical node with largest rank that satisfied every constraints and the node is unmapped with any other nodes from the virtual network requests
5:   if failed to map virtual controller  $v_i$  then
6:     return FAILURE
7:   end if
8: end for
9: return CTR_EMBEDDING_SUCCESS

```

Algorithm 2: Vswitch Embedding Algorithm Based on Degree and Closeness

```

1: DegreeandClosenessRank( $G_p$ )
2: mapping vswitch into the current physical cloud datacenter network
3: waiting until  $G_{vi}$  arrives
4: DegreeandClosenessRank( $G_v$ )
5: for all the unembedded virtual switch nodes do
6:   choose the virtual switch node  $v_i$  with largest degree and closeness
7:   mapping  $v_i$  to the physical network that
8:     a. unmapped
9:     b. satisfy the requirement of  $v_i$ 
10:    c. with the largest  $R_{dc}$ 
11:   if failed to map  $v_i$  then
12:     return FAILURE
13:   end if
14: end for
15: return VSWITCH_EMBEDDING_SUCCESS

```

Algorithm 3: Virtual Path Embedding Algorithm

```

1: for all requests that node mapped successfully do
2:   for each virtual controller-to-switch do
3:     searching the k-shortest paths for increasing k from  $n_s$  to  $n_x$  and
       label the path to the disjoint path
4:     if find the disjoint path then
5:       mapping the shortest paths of the requests to physical link
       according the searching results
6:     end if
7:   end for
8:   generating r commodities for residual edges
9:   if (multicommodity_flow( $v_{vi}$ ,  $v_{vj}$ )) then
10:    return 0
11:  else
12:    find the bottleneck substrate link
13:    randomly choose one virtual link that mapped at the bottleneck
       link
14:    mapping one virtual node of the virtual link into another sub-
       strate node with maximum residual resource  $R_{dc}$  defined
       formula (11)
15:    (multicommodity_flow( $v_{vi}$ ,  $v_{vj}$ ))
16:  end if
17: end for

```

VNR will be placed in the queue only if there are existing any one virtual switch node that does not satisfy the request (lines 11–13). To attempt repeatedly to map the virtual network request in the time window, the virtual network request is mapped successfully just that all virtual switch nodes are mapped successfully (line 15). Otherwise, it will discard the virtual network request at this time.

The mapping of virtual link is shown in algorithm 3. Once the virtual nodes (i.e., controllers and switches) of VNR is mapped successful, the virtual link will try to be mapped following the great results (line 1). Firstly, the virtual link of controller-to-switch is searched by using k-shortest paths algorithm and the virtual controller-to-switch link is successfully mapped into the disjoint path (or this request should not be accepted) (lines 2–7). Secondly, the process of disjoint path method reduces the chance of the path to be embedded successfully because of the constraints. In order to improve the usage of bandwidth, the left virtual links take use of multi-commodity path to improve the acceptance ratio except for vcontroller-to-vswitch path. For flexible path splitting, virtual link embedding problem can be reduced to the Multicommodity Flow Problem (MFP) [17], which can be solved in polynomial time, and $\text{multicommodity_flow}(v_{vi}, v_{vj})$ is defined as the path flow function between virtual node v_{vi} and virtual node v_{vj} . The process is as follows: generating a set of r commodity for a set of virtual link of the residual links of virtual network request, the algorithm will attempt to find all paths for commodity r (lines 8–11). However, because of resource constraints, the virtual link embedding also may be fail even using multipath method. We make fully use of node remapping strategy for multi-commodity approach, which further improves the success ratio of the virtual link embedding (lines 12–15).

6. Experimental Evaluation

This section compares TSVNE algorithm with other similar algorithms to analyse the probability of connective loss, network delay, acceptance rate of virtual network requests and its performance. Firstly, this paper introduces the experimental environment and implements BL, A-NCM and TSVNE algorithms based on the following experimental environment. After that we compare the network connectivity, network delay, runtime overhead, the success acceptance ratio and long-term Rev/Cos of the virtual network requests by the experimental data analysis. It is shown that TSVNE algorithm has achieved more survivable features than BL and A-NCM without using any backup resources and almost no degrading in embedding efficiency, acceptance rate and Rev/Cos ratio.

6.1 Experimental Setup

We deployed a virtualization environment through VMware® Workstation 12 Pro. Ubuntu 14.10 (3.16.0-44-generic) hosted on Windows 7 64 bit© 2009 Microsoft Corporation is allocated 1 core and 4Gb RAM and running on a physical server which is configured with Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz and 8.00GB RAM.

In order to illustrate features of the approach, the regular physical datacenter network is composed of 100 nodes and 570 links, which is generated by GT-ITM [21] tool. The setup is same to our previous work TMVCE [18] and adds the network delay variables. The physical link delay of cloud datacenter network follows uniform distribution from 0ms to 20ms and virtual links are real numbers uniformly distributed between 0 and 50. CPU and bandwidth resources of the physical datacenter network are equivalent to medium-sized network environment and follow uniform distribution from 50 to 100. And then A-NCM and TSVNE algorithm are implemented based on a modification of the open source embed [22] and the actual effect is evaluated through a large number of virtual network requests. We assume that the arrivals of virtual network requests follow the Poisson distribution [23] with an average arrival rate of 5 VNRs per 100 time units, and each has an exponentially distributed lifetime with an average of 1000 time units. In order to fairly compare with the similar algorithms, the balance factor α and β is 1:1, which is similar to [17] and means that CPU and bandwidth have a same important degree.

The network connectivity of virtual network based the environment which is composed of Mininet and OpenVirtex is evaluated through different network topology, for example, the star, ring and tree-like topology. Then, we further evaluate the impact on the survivability of virtual network by the comparison of the different scale topology and three embedding algorithm to analyse the influence factors respectively. Meanwhile, there are three important things to mention. Firstly, it is impractical to cover all failure causes, in general, that affect the states of control paths.

Table 2 Algorithm representation

Notation	Description
BL	Baseline Algorithm including node/edge mapping [17]
A-NCM	Virtual Topology Mapping in SDN-enabled Clouds [12]
TSVNE	Topology-aware survivable virtual network embedding algorithm [our proposed]

In practice, the possibility of multiple physical components fail simultaneously in one physical datacenter network domain is extremely small, and most of the failures only involve single network component in IP networks according to [24]. Therefore, this paper only analyzes the failure scenarios assuming at most one physical component fails at any time in this work. Secondly, the failures of different control paths may be independent and two control paths of different virtual network may share the same physical node or link, and thus control paths fail in a dependent model. Several dependent failure models have been proposed in the past research about network reliability. This paper adopts a commonly-used dependent failure model, the cause-based reliability analysis model [25], in SDN control network. The basic idea is that failures of the control network components, for example, control paths, have underlying physical causes that can be explicitly identified and are statistically independent. Thirdly, any control path protection mechanisms in this work is not adopted. In other word, when a physical component fails, the control paths traversing the physical component are assumed to fail as well.

For each virtual network request, the number of virtual nodes is determined by a uniform distribution between 5 and 50 and the average virtual network connectivity is fixed at 50% that means the physical network node has $n(n-1)/4$ links on average, which is similar to our work [18]. The requirements of the CPU and bandwidth of virtual network requests are real numbers uniformly distributed between 20 and 40. The experiments have fully analysed BL, A-NCM and TSVNE algorithm, and the notations are presented in Table 2.

6.2 Experimental Results

The connectivity of controller-to-switch is extremely vital for the network survivability. There is divergent failure probability for different network topology because of the physical link loss. And we assume each physical node only can hold one virtual controller node. Figure 2 shows that virtual controller connectivity changes with the failure probability of physical link. For this three traditional network topologies, the star topology has the better connectivity because of the smaller network path length. This experiment show the path length or the number of hops, which can be the metrics of network delay, between controller and the switch is an important factor for the network topology. It can be seen that the failure of network components is serious related to the survivability of virtual network.

Physical topology reflects the change of network scale and network centrality. In order to evaluate the availabil-

Fig. 2 Probability of controller connectivity loss **Fig. 3** Comparison on survivability with diff. Scale topology

ity of the method, we take use of three scale topology: Big Scale (500 nodes and 2400 links), Regular Scale (100 nodes and 570 links) and Small Scale (50 nodes and 160 links), and the process is same to above description. Figure 3 shows that TSVNE adapts the change of physical topology, while Big Scale has a small influence of the connectivity loss. This is because TSVNE is closely relative to the selection of virtual SDN controller and network centrality rather than the number of physical components according to the Eqs. (1) and (13).

The vcontroller embedding of TSVNE in this paper has large impact on connectivity. We also use simulation to verify the benefit of our embedding algorithm in Fig. 4. The results show that a clear benefit of TSVNE algorithm in our proposed scheme has achieved compared to A-NCM and BL algorithm. It is observed up to 7% improvements with BL based algorithm, when the failure probability $p = 0.1$ is becoming larger. Even if $p = 0.01$, the improvement of TSVNE is 4% compared to the BL and 2% compared to A-

Fig. 4 Comparison on survivability with diff. Embedding algorithm

Fig. 6 Comparison of controller-to-switch delay with scales

Fig. 5 Comparison of controller-to-switch delay with time

Fig. 7 Comparison of virtual network acceptance ratio with scale

Fig. 8 Comparison of virtual network acceptance ratio over time

NCM.

Taking into account the susceptible network delay of controller-to-switch, virtual link embedding of controller-to-switch adopts unsplitable k-shortest method to reduce the extra long-term average controller-to-switch delay, which is beneficial to approximate the actual product environment. As shown in Fig. 5, the network latency is slightly better than A-NCM, while has a near 4 times delay optimization compared to BL, because TSVNE takes into account the network centrality with the closeness and betweenness attributes. Meanwhile, we design three different scales of VNRs for the pressure test, which is small, regular and large sizes respectively. And the number of nodes is uniform distribution between 5 and 10, 10 and 20, 20 and 50, respectively. It is shown in Fig. 6 that the delay of TSVNE is not increasing faster than other algorithms with the VNRs scale extending. Figure 7 shows that the acceptance ratio of TSVNE is not decreasing quicker than other two algorithms with test pressure increasing. And it is shown in Fig. 8 that the acceptance of virtual network request is closely relation

Fig. 9 Comparison of three algorithm Rev/Cos ratio

Fig. 10 Comparison of three algorithm processing time

with the optimized node ranking, which leads to higher acceptance ratio but almost no constraints of the survivable virtual controller mapping. TSVNE has taken a little advantage in acceptance ratio that is compared with A-NCM but close to 10% higher than BL, which is shown in Fig. 8. Clearly, it manifest that there is little influence on the acceptance ratio of virtual network embedding for a priori selection placement for the virtual controller nodes.

TSVNE can get more revenue than two other algorithms and the features of network survivability are almost no impact on the profits of virtual network embedding, which is shown in Fig. 9. It is the main reason that the selection process of the virtual nodes is optimized and the number of remapping caused by unsuitable virtual network embedding is decreased by our proposed algorithm.

The embedding runtime reflects the waiting time of the cloud tenants and has an important influence on the tenant's experience for the virtual network request in the cloud dat-

acenter. So that this paper optimizes the efficiency and decreases the runtime of virtual network embedding. Firstly, VNR size (n) is defined to the scales of VNRs and n is the number of virtual nodes, which is shown in Fig. 10, and the experiments illustrate that TSVNE consumes less runtime of virtual network embedding compared to the other two similar algorithms. It is the main reason that our proposed approach improves the nodes and links embedding of virtual network request based on degree and closeness as Eq. (11).

7. Conclusion

Delivering virtual network is a cost-effective way for NaaS in cloud datacenter. Survivable virtual network embedding is an enhanced approach to implement multiple virtual network isolation and network flow control, which is beneficial to the virtual network innovation and can provide a pay-as-you-go way for multi-tenants based on cloud computing. Thus, this paper develops a topology-aware survivable virtual network embedding that combines the degree and closeness factors, which is closely associated with the network topology, with the survivable factor, which is used to optimize the survivability of virtual network and improve users' experience, to optimize virtual network reliability. It is shown that our proposed algorithm can effectively enhance virtual network survivability and almost without degrading the performance and acceptance ratio.

In the future, some other survivability of virtual network will be explored to further improve the virtual network reliability. On the other hand, the network traffic congestion of virtual network will be another goal of our research to improve the experience.

Acknowledgments

We would like to thank the project of (2016YFB0800800) the National Key Research and Development Program of China and (2016ZX01040101) Mobile Terminal Operating System for the financial support.

References

- [1] R. Sherwood, "FlowVisor: a network virtualization layer," Online available: <http://www.openflow.org/downloads/technicalreports/openflow-tr-2009-1-flowvisor.pdf>
- [2] A. Al-Shabibi, M. De Leenheer, M. Gerola, A. Koshibe, G. Parulkar, E. Salvadori, and B. Snow, "OpenVirtX: Make your virtual SDNs programmable," In *ACM SIGCOMM HotSDN Workshop*, pp.25–30, 2014.
- [3] P. Costa, M. Miglavacca, P. Pietzuch, and A. Wolf, NaaS: network-as-a-service in the cloud, 2012 USENIX: Management of Internet, Cloud, and Enterprise Networks and Services (Hot-ICE'12), San Jose CA, April 2012.
- [4] Y. Zhu and M. Ammar, "Algorithms for Assigning Substrate Network Resources to Virtual Network Components," in *Proc. IEEE INFOCOM*, vol.2, pp.1–12, 2006.
- [5] 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.

- [6] Z. Wang, J. Wu, Y. Wang, N. Qi, and J. Lan, "Survivable virtual network mapping using optimal backup topology in virtualized SDN," *China Commun.*, vol.11, no.2, pp.26–37, 2014.
- [7] X. Li, H. Wang, B. Ding, X. Li, and D. Feng, "Resource allocation with multi-factor node ranking in data center networks," *Future Generation Computer Systems*, vol.32, pp.1–12, 2014.
- [8] M.R. Rahman and R. Boutaba, "SVNE: Survivable Virtual Network Embedding Algorithms for Network Virtualization," *IEEE Transactions on Network and Service Management (TNSM)*, vol.10, no.2, pp.105–118, 2013.
- [9] R.R. Oliveira, L.R. Bays, D.S. Marcon, M.C. Neves, L.S. Buriol, L.P. Gasparly, and M.P. Barcellos, "Dos-resilient virtual networks through multipath embedding and opportunistic recovery," *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, ACM, pp.597–602, 2013.
- [10] M.G. Rabbani, M.F. Zhani, and R. Boutaba, "On achieving high survivability in virtualized data centers," *IEICE Trans. Commun.*, vol.E97-B, pp.10–18, 2014.
- [11] E. Salvadori, R.D. Corin, A. Broglio, and M. Gerola, "Generalizing Virtual Network Topologies in OpenFlow-Based Networks," *IEEE Global Conference*, Houston, Dec. 2011.
- [12] C. Papagianni, G. Androulidakis, and S. Papavassiliou, "Virtual topology mapping in SDN-enabled clouds," *NCCA* 2014.
- [13] S. Luo, H. Yu, L. Li, D. Liao, and G. Sun, "Traffic-Aware VDC Embedding in Data Center: A Case Study of FatTree," *China Communications*, vol.11, no.7, pp.142–152, July 2014.
- [14] S.-Q. Gong, J. Chen, Q.-Y. Kang, et al., "Efficient and coordinated mapping algorithm in virtualized SDN networks," *Front. Inform. Technol. Electron. Eng.*, vol.17, no.7, pp.701–716, 2016.
- [15] R.L. Gomes, L.F. Bittencourt, E.R.M. Madeira, E. Cerqueira, and M. Gerla, "Bandwidth-aware allocation of resilient virtual software-defined networks," *Comput Netw*, vol.100, no.5, pp.179–194, 2016.
- [16] R. Li, J. Zhang, Y. Tan, and Q. Wu, "Topology-Aware VSDN Embedding Approach On Resilient Software Defined Networks," *5th International Conference on Computer Science and Network Technology (ICCSNT)*, 2016.
- [17] M. Yu, Y. Yi, J. Rexford, and M. Chiang, "Rethinking Virtual Network Embedding: Substrate Support for Path Splitting and Migration," *ACM SIGCOMM Comput. Commun. Rev.*, vol.38, no.2, pp.17–29, April 2008.
- [18] R. Li, J. Zhang, Y. Tan, and Q. Wu, "TMVCE—topology-aware multipath Virtual Cluster embedding algorithm," *Proc. Int. Conf. Cloud Comput. Internet Things, CCIOT*, pp.57–63, 2016.
- [19] Z. Wang, Y. Han, T. Lin, H. Tang, and S. Ci, "Virtual network embedding by exploiting topological information," *Proc. IEEE GLOBECOM*, pp.2603–2608, 2012.
- [20] X. Cheng, S. Su, Z. Zhang, H. Wang, F. Yang, Y. Luo, and J. Wang, "Virtual network embedding through topology-aware node ranking," *SIGCOMM Comput. Commun. Rev.*, vol.41, no.2, pp.38–47, April 2011.
- [21] E.W. Zegura, K.L. Calvert, and S. Bhattacharjee, "How to Model an Internetwork," in *Proc. 15th Annu. Joint Conf. IEEE INFOCOM*, vol.2, pp.594–602, 1996.
- [22] <https://github.com/USC-NSL/embed.git>
- [23] G. Wang, J. Li, and X. Chang, "Modeling and performance analysis of the multiple controllers' approach in software defined networking," *IEEE 23rd International Symposium on Quality of Service (IWQoS)*, pp.73–74, 2015.
- [24] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C.-N. Chuah, Y. Ganjali, and C. Diot, "Characterization of failures in an operational ip backbone network," *IEEE/ACM Trans. Netw.*, vol.16, no.4, pp.749–762, Aug. 2008.
- [25] K.V. Le and V.O.K. Li, "Modeling and analysis of systems with multimode components and dependent failures," *IEEE Trans. Rel.*, vol.38, no.1, pp.68–75, April 1989.
- [26] D.-G. Cao, B. An, P.-C. Shi, and H.-M. Wang, "Providing virtual cloud for special purposes on demand in JointCloud computing environment," *J. Comput. Sci. Technol.*, vol.32, no.2, pp.211–218, March 2017.

- [27] L.F. Muller, R.R. Oliveira, M.C. Luizelli, L.P. Gasparly, and M.P. Barcellos, "Survivor: An enhanced controller placement strategy for improving SDN survivability," *global communications conference*, pp.1909–1915, 2014.

Rongzhen Li received the Master degree in 2013, both from National University of Defense Technology. He is currently a Ph.D. student of National University of Defense Technology. His research interests include parallel and distributed system and cloud computing.

Qingbo Wu received the Ph.D. degree in computer science and technology from National University of Defense Technology in 2010. Now he is a professor and master supervisor at National University of Defense Technology. His research interests include operating system and cloud computing.

Yusong Tan received the Ph.D. degree in computer science and technology from National University of Defense Technology in 2004. Now he is a professor and master supervisor at National University of Defense Technology. His research interests include cloud computing and big data.

Junyang Zhang received the Master degree in 2013, both from National University of Defense Technology. He is currently a Ph.D. student of National University of Defense Technology. His research interests include Artificial Intelligence and Deep Learning.