

An Efficient VNE Algorithm Via Preferentially Mapping Important Nodes

Yanyu Li, Yunchun Li
Sino-German Joint Software Institute,
School of Computer Science and Engineering
Beihang University, Beijing, China
Email:lyy120@buaa.edu.cn ; lych@buaa.edu.cn

Min Shu, Jiqiang Tang, Yigang Peng
National Computer Network Emergency
Response Technical Team
Coordination Center (CNCERT/CC)
Email:tangjq@cert.org.cn

Abstract—Virtual Network Embedding (VNE) is one of the most challenging problems to realize network virtualization. In this paper, we propose that the method of sequencing virtual nodes can affect VNE algorithms' efficiency. Then we propose a novel strategy to coordinate the node embedding and link embedding. Based on the semi-local centrality measure used in social network, our strategy can present the importance of node. It can be declared as the sum of a virtual node's resource capacity and its neighbor nodes' resources capacities (SR). Furthermore, we design an efficient heuristic VNE algorithm, which is called SR-VNE. Extensive simulation experiments show that the proposed SR-VNE algorithm more effectively improves the time-average revenue as well as the acceptance ratio compared to the existing VNE algorithms.

Index Terms—Network Virtualization, Virtual Network Embedding (VNE), Combination of Local Topology and Resources

I. INTRODUCTION

Recently, network virtualization has gained intensive attention because of its promising solution for the Internet's ossification [7]. At the same time, network virtualization has been widely used in Cloud Computing and Data Center, such as projects [2, 9, 10]. With the use of network virtualization, multiple heterogeneous virtual networks (VNs) are allowed to coexist on the same shared substrate network (SN). In this case, the conventional Internet service provider (ISP) can be decoupled into infrastructure providers (InPs) and service providers (SPs). Infrastructure providers (InPs) maintain the shared substrate network (SN), which is composed of substrate nodes and links. Service providers (SPs) lease physical resources (such as CPU, bandwidth and so on) from InPs and create customized virtual networks (VNs) to provide services for end users. However, how to embed multiple virtual networks with resource constraints into a substrate network is one of the fundamental challenges of network virtualization, which is known as the virtual network embedding problem (VNE). Our research is used for the experimental platform which is mixed with virtualization and non-virtualization. The platform is used as the InPs and SPs for users.

The VNE problem has been certified to be NP hard [1]. A number of researches have been proposed [4–6, 8, 11–15, 17–19] in VNE problem. Lischka et al. [12] proposed a backtracking Virtual Network Embedding algorithm based

on subgraph isomorphism detection: it maps nodes and links during the same stage. While others tackle the VNE problem in two separate stages [4–6, 8, 11, 13–15, 17–19]: nodes embedding stage and links embedding stage. They usually apply greedy strategy to select virtual nodes and available substrate nodes. In link embedding phase, one way uses shortest paths strategy when paths are not allowed to split or migrate. The other way transforms link embedding problem as the multicommodity flow problem when paths are able to split. In addition, Chowdhury et al. [8] firstly proposed two coordinated VNE algorithms which make coordination between the node mapping and the link mapping. The authors [5, 11, 14, 17] also proposed their algorithms to make better coordination between nodes mapping and links mapping. However, what is the key point to optimize VNE algorithms?

In this paper, we would tackle the VNE problem in a different perspective. Firstly, we propose a novel strategy to coordinate the node embedding and link embedding processes efficiently. Based on the semi-local centrality measure [3] used to calculate the influence of nodes in social network, we use the node's local topology attribute and resources to reflect its importance. Our metric can be defined as the sum of a virtual node's rank and its neighbor nodes' ranks (shorted as SR, sum of rank). In this paper, we only consider each node's nearest neighbor nodes' influence on it. Each node's rank is calculated as the resources capacity metric in paper [15]. Secondly, we only use SR to sort the virtual nodes because SR can not reflect node's resource capacity. What's more, we make experiments to verify the different existing metrics to calculate substrate nodes' resource in [5, 11, 15] can return similar results. We use the metric in [15] to reflect substrate nodes' resources capacity. Thirdly, we propose a coordinated two stages virtual network embedding algorithm based on SR, which we call it as SR-VNE algorithm (sum of ranks virtual network embedding). We sort the substrate nodes in decreasing order with its ranking value [15]. And we determine the sequence of virtual nodes of the VN request in non-increasing order according to the topology-aware SR metric. As we found that the virtual node with many resource demands and large degrees is easy to be bottleneck node from a large number of previous experiments, we preferentially embed the most important virtual nodes. We use greedy method in the node

embedding phase. The node embedding algorithm can embed the most important virtual node onto the substrate node with largest available resources. After the node embedding succeed, the SR-VNE algorithm leverages the shortest-path method to embed virtual links. Numerical results demonstrate that our algorithm more effectively improves the acceptance ratio and the long-time average revenue than the algorithms [5, 11] which also consider the topology attribute of the network.

The rest of this paper is organized as follows. We discuss the related work in Section II. The virtual network embedding problem is described in Section III. In Section IV, we present our thinking and method of determining the importance of virtual nodes. Then our SR-VNE algorithm is proposed in Section V. The VNE algorithms are evaluated in Section VI. Section VII summarizes the paper.

II. RELATED WORK

Virtual network Embedding algorithms can be divided into two categories: One-stage VNE algorithms and two-stages VNE algorithms. One-stage VNE algorithms tackle node embedding and link embedding in only one stage by using backtracking strategy, which is based on subgraph isomorphism detection [12]. If the substrate network has available embedding results of the waiting VN request, the one-stage VNE algorithm will return one of the results. However, the cost time is too large, especially for big networks. Two-stages VNE algorithms tackle node embedding and link embedding in two stages. Firstly, they use greedy method to achieve node embedding. Then they tackle link embedding with shortest path routing algorithm or path splitting and migration [5, 7, 15]. However, this kind of algorithms can not return good results sometimes. Which means that the SN is available for embedding VN request, the result of the two stages algorithm is rejected. In most cases, the failure is caused by the failure of link embedding phase because there is no path between the two selected substrate nodes. A few of researches [5, 11, 14, 17] propose topology-aware strategy to make coordination between node embedding and link embedding.

Cheng et al. [5] apply the Markov Random Walk model to rank a network node based on its resource and topological attributes. They use the idea of PageRank to measure topology-aware resource ranking of a node, which can reflect the resource and quality of connections of a node. In paper [11], they introduce seven topological characteristics complementary to each other, which measure the relative influence or importance of nodes in the substrate network and VNs from different aspect. Their algorithms use the degree to measure a node's resource capacities. Zhang et al. [17] propose Opportunistic Resource Sharing and Topology-Aware node ranking VNE algorithm. They focus on VNE problem with the position that time-varying resource requirements of virtual networks should be taken into consideration. Gong et al. [14] propose a novel metric, which takes the resources of the entire network into consideration (GRC) to quantify the embedding potential of each substrate node.

Our work differs from the existing algorithms [5, 11, 14, 17] in different perspectives. They all focus on making good use of the infrastructure resource and addressing the big island problem. But what is the most important part that makes these algorithms work better than the algorithm of [15]? Is those metrics can better reflect nodes' resources capacities or reduce the possibilities of bottleneck nodes? The metric GRC [14] can reflect the potential of each node, but it costs to update the information of the SN if the SN is changable. In this paper, we analyze these algorithms' main functions in a novel way to define which part is more important to improve the efficiency of embedding virtual networks [5, 11]. Then we propose a coordinated virtual network embedding algorithm to reduce the bottleneck nodes of VN requests to get better results.

III. PROBLEM STATEMENT AND FORMULATION

In this section, we make a description of the VNE problem, which includes network models, VNE process and VNE objectives.

A. Network models

1) *Substrate Network*: A substrate network can be modeled as a weighted undirected graph $G_s = (N_s, L_s, A_s^n, A_s^l)$, where N_s and L_s represent to the set of nodes and links respectively. The notation A_s^n denotes the attributes of the substrate nodes (capacity, storage, location et al.) and A_s^l denotes the attributes of the links (bandwidth). In this paper, we only consider the available CPU capacity of nodes and the bandwidth of the links.

2) *Virtual Network Request*: Similar to the substrate network, an undirected graph $G_v = (N_v, L_v, A_v^n, A_v^l)$ can be used to represent the virtual network request. N_v is the set of virtual nodes and L_v is the set of virtual links. Then we use A_v^n and A_v^l to denote the attributes of the nodes and links. In this research, we assume that VN request arrival process is a Poisson process [14], with a time-invariant rate of λ . Each VN request has two time-domain parameters: t (the arrival time) and t_l (the lifetime of the VN request).

B. VNE Process

The VNE problem is defined as a mapping M from G_v to a substrate G_s , the mapping $M : G_v(N_v, L_v) \rightarrow G_s(N'_s, L'_s)$, where $N'_s \subset N_s$, $L'_s \subset L_s$. The VNE mapping includes two steps which can be called as node mapping phase and link mapping phase. In node mapping phase, the virtual node's CPU capacity should be satisfied, which can be denoted as $A_v^n(\text{of } n_v) \leq A_s^n(\text{of } n_s)$, $n_v \in N_v$, $n_s \in N'_s$. In link mapping phase, the virtual link's bandwidth is not larger than the selected substrate link's bandwidth: which can be denoted as $A_v^l(\text{of } l_v) \leq A_s^l(\text{of } l_s)$, $l_v \in L_v$, $l_s \in L'_s$. The virtual network request would be rejected or postponed if there are not sufficient substrate resources available in the SN.

What is more, nodes of the same virtual network request can not be embedded on the same substrate node. We ignore this restriction when we deal with different virtual network requests. In this paper, we only consider the link embedding without path splitting and migration.

C. VNE Objectives

The main objective of VNE is to propose an efficient embedding algorithm for the online problem, where VN request arrive and depart over time. From the InPs' perspective, the object of an online embedding algorithm would be to maximize the revenue. Similar to the previous work in paper [8, 15, 19], the revenue of accepting a VN request at time t can be formulated by:

$$R(G_v(t)) = \sum_{n_v \in N_v} CPU(n_v) + \sum_{l_v \in L_v} BW(l_v) \quad (1)$$

The cost of accepting a VN request at time t is defined as:

$$C(G_v(t)) = \sum_{n_v \in N_v} CPU(n_v) + \sum_{l_v \in L_v} |H(l_v^s)| BW(l_v) \quad (2)$$

Where $|H(l_v^s)|$ is the length of path $l_v^s(l_v \rightarrow l_v^s)$ in terms of hops.

Our objective is to maximize the long-term average revenue, which is given by

$$\lim_{T \rightarrow \infty} \frac{\sum_{t=0}^T R(G_v(t))}{T} \quad (3)$$

In addition, we also consider the acceptance ratio of the substrate network from the SPs' perspective. Obviously, the acceptance ratio is related to the metric (3), it can be defined as

$$\lim_{T \rightarrow \infty} \frac{\sum_{t=0}^T N(VNR_s)}{N(VNR)} \quad (4)$$

Where $N(VNR)$ denotes the number of all VN requests, $N(VNR_s)$ is the number of VN requests successfully accepted by the substrate network. What's more, the VN request Revenue-to-Cost Ratio is also related to the metric (3), it is defined as

$$R/C = \lim_{T \rightarrow \infty} \frac{\sum_{t=0}^T R(G_v(t))}{\sum_{t=0}^T C(G_v(t))} \quad (5)$$

We prefer to compare VNE algorithms in terms of the long-term average revenues firstly. If the long-term average revenues are about the same, the higher acceptance ratio is preferred. Then we would check the R/C ratio.

IV. METHOD DESCRIPTION

When we studied the algorithms of two-stages VNE algorithms, we found a problem. The problem of this kind of algorithms can be described as: the substrate network can embed the VN request in reality, but the algorithm may not return right reason because of the failure of link embedding. We think that's the point why researchers paid effort on making good coordination between node embedding process and link embedding process [5, 11, 14, 17]. They all consider the topology attributes of the network, and use topology attributes to calculate a node's resources capacity.

In this paper, we tackle the VNE problem in different perspective. As we all know, a virtual node with large degree is easy to be a bottleneck node in the link embedding phase, especially its nearest neighbor nodes' resources requirements

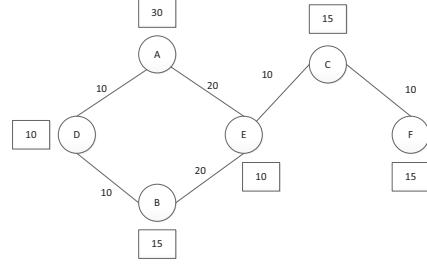


Fig. 1: A simple network

are big. This kind of nodes can be called as important nodes in Graph Theory. What we want to do is to propose a algorithm to handle this kind of virtual nodes efficiently. Luckily, we learn a lot from semi-local centrality measure [3], which has been widely used in social network. The semi-local centrality measure can reflect a node's importance in network. When it calculates a node's influence, it calculates the sum of the node's degree, its nearest neighbor nodes' degree and its second-nearest neighbor nodes' degree. Based on semi-local centrality measure, we propose a strategy to calculate a node's importance in VNE problem. We agree with that a node's neighbor nodes have impact on it, and we only consider the nearest neighbor nodes for brief.

Firstly, we use the ranking metric in Equation (6) proposed in paper [15] to reflect a node's resource capacity. Secondly, we calculate the sum of a node's ranking value and its nearest neighbor nodes' ranking values. The sum of ranking value can be shorted as SR (sum of ranks), which is denoted in Equation (7). What's more, the importance of nodes can be calculated with their SR values. In our opinion, the node with the largest SR value is the most important node in network. If every node's SR is the same, we use the ranking metric in Equation (6) to calculate nodes' importance. Equation (6) and Equation (7) are defined as the follows:

$$H(n) = CPU(n) \sum_{l \in L} BW(l) \quad (6)$$

Where $CPU(n)$ is the CPU resource of node n (remaining resource for substrate node, capacity constraints for virtual node), $BW(l)$ is the bandwidth of link l .

$$SR(n) = H(n) + \sum_{u \in Nbr(n)} H(u) \quad (7)$$

Where $Nbr(n)$ is denoted as the nearest neighbor nodes of node n .

However, if a node is the most important node in the network, its available resources capacity might not be the largest. As we give an example in Fig. 1, if we use Equation (6) to calculate available resource capacity, $H(a) = 30 * (10 + 20) = 900$, $H(b) = 450$, $H(c) = 300$, $H(d) = 200$, $H(e) = 500$, $H(f) = 150$. Use the Equation (7), we can see that $SR(a) = H(a) + (H(d) + H(e)) = 900 + (200 + 500) =$

1600, $SR(b) = 1150$, $SR(c) = 950$, $SR(d) = 1550$, $SR(e) = 2150$, $SR(f) = 450$.

As the Fig. 1 shows, we can see that the node e is the most important node, but node a is the node with the largest available resources. The degree of node a, node b, node c, node d is same, then the resource capacity of itself and neighbor nodes can impact the node's importance. If we want to calculate which one node between two or more nodes is more importance than the other one or others, we would consider different cases. For example, node a and node b have the same nearest neighbor nodes, that means the sum of their neighbor nodes' resources is same. Then the resource capacities of the two nodes themselves impact the SR values. $H(a) > H(b)$, then $SR(a) > SR(b)$. What's more, although the node d has less resource capacity than node b, node d's neighbor nodes have more resource capacities than node b's neighbor nodes, and the SR value of node d is larger than the SR value of node b. This case can show a node's neighbor nodes' resource capacity can impact the node's importance a lot.

If a node has a larger SR value than any other nodes in network, it can only mean that the node is the most important node in network. The metric can reflect a node's connective relationship and local resources capacity, but not the resources capacity of itself. That's why we only use SR to calculate the importance of virtual nodes in VN requests, and we use the metric in Equation (6) instead of SR to denote the available resources capacity of substrate nodes. Our idea is to embed the most important unembedded virtual node onto the substrate node with the largest available resource capacity. We think it can decrease the bottleneck of link embedding. We will describe our SR-VNE algorithm in detail in Section V.

In addition, we compare the two new algorithms RW-MM-SP [5] and DR-MM-SP [11] and the commonly used algorithm BL-SP [15] to show the ranking metric of Equation (6) is good enough to reflect nodes' resource capacity. We use DR (Degree Rank) to denote ND metric in paper [11] to distinguish with NodeRank. The differences of these three algorithms are described in table I. There are two differences between these three algorithms. Firstly, these three metrics of calculating substrate nodes and virtual nodes are different. Cheng et al. and Feng et al. proposed new topology-aware metric to better coordinated the node embedding and link embedding than the metric of Yu et al. Secondly, two new algorithms [5, 11] use their metric to sort the virtual nodes and substrate nodes respectively. However, BL-SP algorithm [15] just use their metric to sort the substrate nodes. Then we propose SRW-SP algorithm and SDR-SP algorithm to find out the reason why these algorithms work better than algorithm in paper [15]. SRW-SP and SDR-SP both contain two embedding stages (node embedding and link embedding). SRW-SP use the metric in paper [5] and SDR-SP use the metric in [11] to sort virtual nodes. SRW-SP and SDR-SP both use the metric in paper [15] to sort substrate nodes. The details can be seen in Table II. Then we compare SRW-SP to RW-MM-SP, compare SDR-SP to DR-MM-SP. What's more, in this paper, we only consider the link embedding used shortest-path method. The

TABLE I: The Difference between the algorithms[5, 11, 15]

Difference	BL-SP[15]	RW-MM-SP[5]	DR-MM-SP[11]
1. Sort the virtual nodes	—	With their NodeRank metric	With their DR metric
2. Sort the substrate nodes	With their ranking metric	With their NodeRank metric	With their DR metric

experiment results will show in Fig. 2 and Fig. 3.

V. SR-VNE ALGORITHM

Our SR-VNE algorithm is a two-stage VN embedding algorithm, it contains Algorithm 1 (Node Embedding Algorithm) and Algorithm 2 (Link Embedding Algorithm). In the node embedding phase, we use the ranking metric [15] denoted in Equation (6) and our SR metric denoted in Equation (7) to calculate the virtual nodes' sum of rank. As we declare in Section IV, we want to embed the most important virtual nodes onto the substrate node with most resources capacity to increase the acceptance ratio and time-average revenues. Then we sort the virtual nodes of the coming VN request in non-increasing order according to their SR values. At the same time, we calculate the substrate nodes' resource capacities according to the metric in Equation (6). Then we sort the substrate nodes in non-increasing order according to their H values. Without loss of generality, let the sorted virtual nodes be

$$N_v^1, N_v^2, \dots, N_v^m, \text{ where } SR_1 \geq SR_2 \geq \dots \geq SR_m$$

and let the sorted substrate nodes be

$$N_s^1, N_s^2, \dots, N_s^n, \text{ where } H_1 \geq H_2 \geq \dots \geq H_n$$

In addition, we use the same sort algorithm in all VNE experiments, so that sort algorithm does not affect the differences of these algorithms' efficiency.

Our strategy points out the most important virtual node should embed on the SN preferentially to reduce the potential to be the bottleneck of the VNE algorithm. We think the ranking metric in Equation (6) is available for describing the resource capacities of nodes. In this way, we would use greedy strategy to embed the most important virtual node (with the largest SR value) onto the substrate node with the largest H value. It then maps N_v^i to N_s^i where $i = 1, 2, \dots, m$. Algorithm 1 shows the details of the SR-VNE node mapping algorithm. If the result is NODE-EMBEDDING-FAILED, it means that the VN request will be rejected.

After embedding all the virtual nodes of VN request, we would use shortest-path algorithm to solve the link embedding problem. Algorithm 2 shows the details of the link-embedding process.

As the SR metric considers the topology attributes of the network, the algorithm would optimize the coordination between node embedding and link embedding phases. We tackle the important virtual nodes at the beginning so that its possibilities to be bottleneck node will be reduced. We would present our experiment results in Section VI.

Algorithm 1 :SR-VNE Node Embedding

Input: : The substrate network G_s ,The waiting virtual network request G_v .

Output: : Node Embedding Results

- 1: Computing the Rank Values(H) of all nodes of VN request G_v and substrate network G_s using the metric (6).
- 2: Computing the Rank of all nodes in G_v with the metric (7).
- 3: Sort the nodes in G_v according to their SR values in non-increasing order
- 4: Sort the nodes in G_s according to their H values in non-increasing order
- 5: Map virtual nodes to substrate nodes with the sequences of their order respectively
- 6: **if** node resource constraints satisfied **then**
- 7: return NODE-EMBEDDING-SUCCESS
- 8: **else**
- 9: return NODE-EMBEDDING-FAILED
- 10: **end if**

Algorithm 2 :Link Embedding based on Shortest-Path

Input: : The substrate network G_s ,The virtual network request G_v ,the node-embedding results

Output: : Link Embedding Results

- 1: for each link $l_v = (n_{v1}, n_{v2})$ in G_v
- 2: use k-shortest path strategy to find a substrate path of (n_{s1}, n_{s2}) ,which $(n_{v1} \rightarrow n_{s1} \text{ and } n_{v2} \rightarrow n_{s2})$
- 3: **if** link resource constraints satisfied **then**
- 4: return LINK-EMBEDDING-SUCCESS
- 5: **else**
- 6: return LINK-EMBEDDING-FAILED
- 7: **end if**

VI. PERFORMANCE EVALUATION

In this section, we present the simulation model at the beginning. Secondly, we present our evaluation results. The results can show that our SR-VNE algorithm gets more time-average revenue as well as the acceptance ratio compared to the two VNE algorithms [5, 11]. In addition, the comparisons of algorithms in [5, 11, 15] verify that it's important to consider the topology attributes of VN requests. At the same time, our results show that the metric of Equation (6) can reflect the available resources capacity of substrate nodes well. That's why we use the Equation (6) to sort the substrate nodes. What's more, we focus on the performance comparison between our SR-VNE algorithm and the two algorithms, the RW-MM-SP algorithm [5] and DR-MM-SP algorithm [11].

A. Simulation Model

We adopt the similar simulation model as previous works [5]. The topologies of substrate network and the VN requests are randomly generated by the GT-ITM tool [16]. The scale of SN and VN requests are as the follows: The node number of the SN is 100, the link number of the SN is 570. The degree

TABLE II: Comparisons of Algorithms

Name of the algorithm	Description of the algorithm
SR-VNE	Equation (7) is used for VN , Equation (6) is used for SN
RW-MM-SP	the metric in paper [5] is used for VN and SN
SRW-SP	VN uses metric in paper [5], Equation (6) is used for SN
DR-MM-SP	the metric in paper [11] is used for VN and SN
SDR- SP	VN uses metric in paper [11], Equation (6) is used for SN

of the SN is from 4 to 20. For each VN request, the number of node is selected from 2 to 20. The connectivity rate (the probability of any two nodes are connected) in a VN request is 0.5. The available computing resources and bandwidth resources of SN are randomly selected (from 50 to 100 units). The computing resources and bandwidth requirement for every VN request are also randomly selected (from 0 to 50 units). The VN requests arrive one by one. The arrival rate of λ requests per time unit, and the lifetime of each request is average of 500 time units. In our experiment, the fixed traffic load is $\lambda*500=5$. We show the comparison algorithms in table II. In addition, the results of Fig. 2 to Fig. 4 are calculated of the average values of eleven times experiments.

B. Evaluation Results

The results of Fig. 2 and Fig. 3 can prove the metric of Equation (6) can reflect the available resources capacity of substrate nodes well. The differences of algorithms [5, 11, 15] show that it's important to sequence virtual nodes to be embedded firstly. Based on this result, we propose our SR-VNE algorithm. Then we compare our topology and resource combination SR-VNE algorithm to previous RW-MM-SP algorithm [5] and DR-MM-SP algorithm [11]. Several performance metrics for evaluation are used as we discuss in Section III, including the time-average revenue defined by Equation (3), the acceptance ratio defined by Equation (4), time-average cost defined by Equation (2), R/C ration defined by Equation (5). We will discuss the details of the results as follows.

1) *The comparison between RW-MM-SP and SRW-SP, DR-MM-SP and SDR-SP respectively:* As we discussed in Section IV, Cheng et al. and Feng et al. proposed new topologyaware metric to better coordinated the node embedding and link embedding than the metric of Yu et al. The differences are that their algorithms use their metric to sort the virtual nodes and substrate nodes respectively. However, Yu et al. just use their metric to sort the substrate nodes. After considering this differences, the SRW-SP algorithm uses NodeRank metric in paper [5] to sort virtual nodes and Equation (6) to sort the substrate nodes. The SDR-SP algorithm use DR metric [11] to sort the virtual nodes and Equation (6) to sort the substrate nodes. We will do experiments to make comparisons between SRW-SP and RW-MM-SP, SDR-SP and DR-MM-SP respectively.

Fig. 2 (a-b) show the results comparison between RW-MM-SP and SRW-SP. As we can see from Fig. 2 (a), the SRW-SP

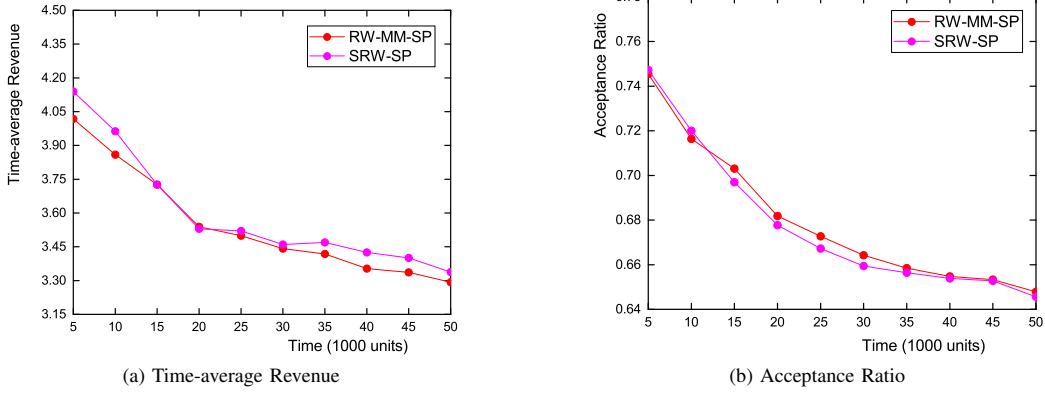


Fig. 2: Comparison of RW-MM-SP and SRW-SP.

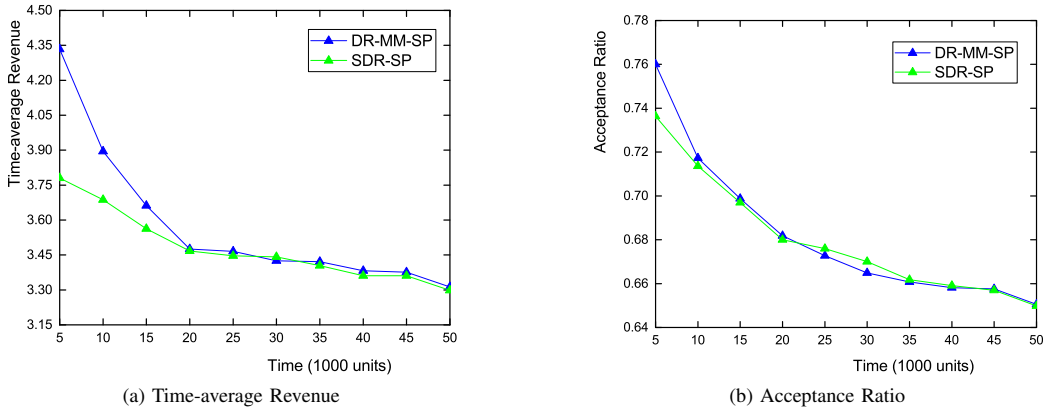


Fig. 3: Comparison of DR-MM-SP and SDR-SP.

algorithm gets more time-average revenue than RW-MM-SP. Fig. 2 (b) shows the nearly same results of the acceptance ratio of the RW-MM-SP and SRW-SP. At the beginning, SRW-SP can receive a little more VN requests than RW-MM-SP. After time 10, the RW-MM-SP overtakes SRW-SP. Compare Fig. 2 (a) with Fig. 2 (b), maybe SRW-SP tends to receive large virtual network requests.

We show the performance comparison between DR-MM-SP algorithm and SDR-SP algorithm in Fig. 3. If you are considering the time-average revenue, you can see the result in Fig. 3 (a). Although at the time from 5 to 20, the time-average revenue of DR-MM-SP is much larger than SDR-SP, the difference of time-average revenue between SDR-SP and DR-MM-SP is decreasing. Fig. 3 (b) shows that the DR-MM-SP rejects more VN request than SDR-SP from time 20 to 50. And the performance difference of these two algorithms is becoming smaller and smaller.

Above all, the results of Fig. 2 and Fig. 3 can show that it's important to sort virtual nodes and consider the topology attributes of VN requests. What's more, it also proves that the metric of Equation (6) can reflect the available resources capacity of substrate nodes as well as NodeRank and DR.

That's why we use the metric of Equation (6) to sort substrate nodes.

2) *The comparison of SR-VNE and RW-MM-SP, DR-MM-SP*: As we can see from Fig. 4 (a), although SR-VNE doesn't work as well as the two algorithms from time 5 to 15, its trend of time-average revenue's changes is the most stable when compares it with RW-MM-SP and DR-MM-SP. Besides, SR-VNE can achieve best time-average revenue from time 20 to 50. What's more, we believe the number between time 20 and 50 is more useful to prove its efficiency than time 5-20. Fig. 4 (b) shows that our topology and resource coordination algorithm SR-VNE can achieve the best acceptance ratio as time goes. DR-MM-SP can achieve the best performance of acceptance ratio at the beginning from time 5 to 10. Then the winner is RW-MM-SP from time 10 to 20. SR-VNE overtakes them at time 20, and it is the best from time 20 to 50. In addition, the SR-VNE is more stable than RW-MM-SP and DR-MM-SP in terms of the acceptance ratio's changes. As we can see from Fig. 4 (c), the cost of SR-VNE is the least from time 5 to 10, then it's the middle one from time 10 to 25 and from time 40 to 50. However, it costs more than the other two algorithms from time 30 to 40. From time 30 to

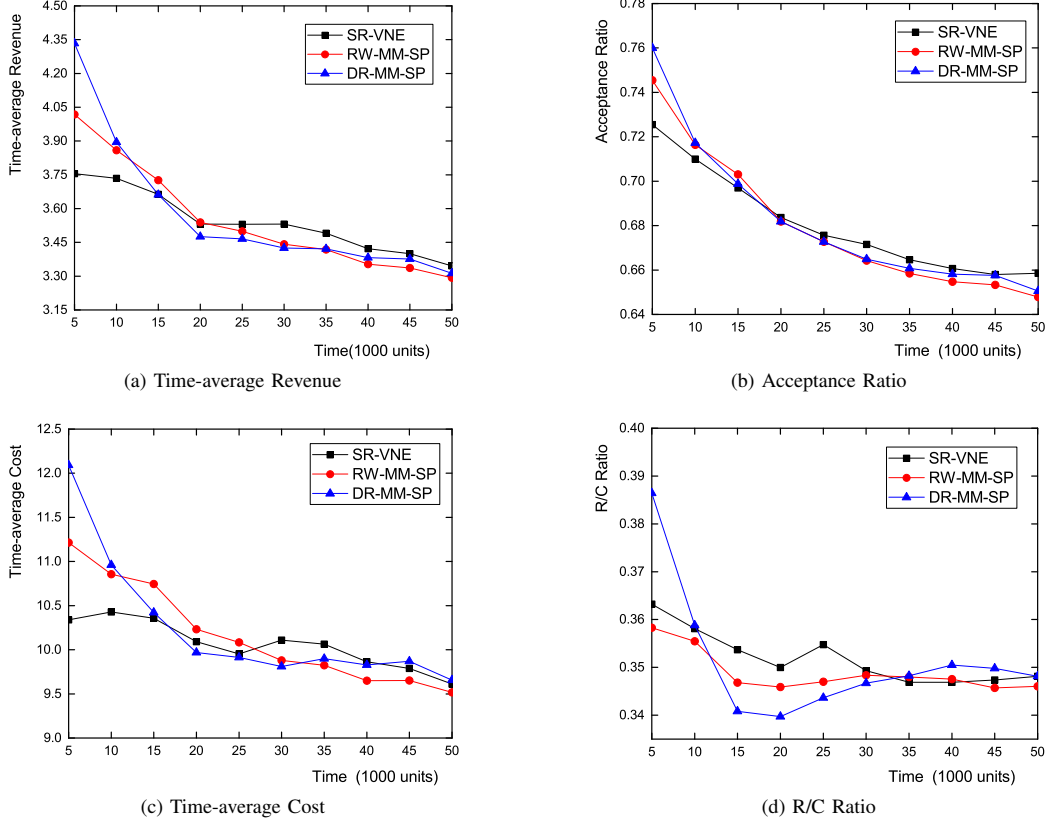


Fig. 4: Comparison of SR-VNE, RW-MM-SP and DR-MM-SP.

40, SR-VNE costs more than RW-MM-SP and DR-MM-SP. As we can see from Fig. 4 (b), SR-VNE receive much more virtual network requests from time 30 to 40, we think it may be the reason that SR-VNE costs a lot. Fig. 4 (d) shows the R/C ratio of these three algorithms. According to the figures, SR-VNE can get best or middle results except between time 35 and 40.

In conclusion, we think SR-VNE can work better than RW-MM-SP and DR-MM-SP, all performance's changes of SR-VNE is more stable than RW-MM-SP's and DR-MM-SP's. From the results of Fig. 4, we can see that SR-VNE can achieve better time-average revenue and acceptance ratio as the traffic load increases, although it may cost more resources to obtain more revenue and acceptance ratio. Besides, the R/C ratio of algorithms can also show SR-VNE is quite better than the other two algorithms.

VII. CONCLUSION

In this paper, we propose a novel topology and resource combination strategy to calculate the importance of node in network. The method (which we call it SR) is based on the semi-local centrality measure of calculating the influence of nodes in social network. Based on this new method, we propose the SR-VNE algorithms. SR-VNE algorithm can coordinates node embedding stage and link embedding stage well,

because the SR metric can reflect the importance of nodes in terms of its resources and connection relationships. We use different metric to calculate virtual nodes and substrate nodes. Firstly, our SR-VNE algorithm embeds the most important virtual node on the substrate network with the most available resources in node embedding phase. Secondly, we use shortest-path strategy to embed link. By using the topology attributes in node embedding phase, the number of bottleneck in link embedding may be solved in some extent.

In the future, we will extend our work to consider a node's k-nearest neighbor nodes in different scale of network. We want to determine which k value can best fit the scale in order to reflect each node's importance ($k > 1$). And maybe we can get a better trade off of the time cost and efficiency.

ACKNOWLEDGEMENT

This work was supported by the National Hitech R&D program of China (863 program) (Grant No.2015AA01A301) and National Science and Technology Major Project of the Ministry of Science and Technology of China (2012BAH45B00).

REFERENCES

- [1] D. G. Andersen, "Theoretical approaches to node assignment," 2002.

- [2] A. Bavier, N. Feamster, M. Huang, L. Peterson, and J. Rexford, "In vini veritas (realistic and controlled network experimentation)," vol. 36, no. 4, pp. 3–14, 2006.
- [3] D. Chen, L. Lu, M. Shang, Y. Zhang, and T. Zhou, "Identifying influential nodes in complex networks," *Physica A-statistical Mechanics and Its Applications*, vol. 391, no. 4, 2012.
- [4] X. Cheng, S. Su, Z. Zhang, K. Shuang, F. Yang, Y. Luo, and J. Wang, "Virtual network embedding through topology awareness and optimization," *Computer Networks*, vol. 56, no. 6, pp. 1797–1813, 2012.
- [5] X. Cheng, S. Su, Z. Zhang, H. Wang, F. Yang, Y. Luo, and J. Wang, "Virtual network embedding through topology-aware node ranking," *Computer Communication Review*, vol. 41, no. 2, pp. 38–47, 2011.
- [6] M. Chowdhury, F. Samuel, and R. Boutaba, "Polyvine (policy-based virtual network embedding across multiple domains)," 2010.
- [7] N. M. M. K. Chowdhury and R. Boutaba, "A survey of network virtualization," *Computer Networks*, vol. 54, no. 5, pp. 862–876, 2010.
- [8] N. Chowdhury, M. R. Rahman, and R. Boutaba, "Virtual network embedding with coordinated node and link mapping," 2009.
- [9] B. Chun, D. E. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman, "Planetlab: an overlay testbed for broad-coverage services," *Computer Communication Review*, 2003.
- [10] N. Feamster, L. Gao, and J. Rexford, "How to lease the internet in your spare time," *Computer Communication Review*, vol. 37, no. 1, 2007.
- [11] M. Feng, J. Liao, J. Wang, S. Qing, and Q. Qi, "Topology-aware virtual network embedding based on multiple characteristics," 2014.
- [12] J. Lischka and H. Karl, "A virtual network mapping algorithm based on subgraph isomorphism detection," 2009.
- [13] S. Qing, Q. Qi, J. Wang, T. Xu, and J. Liao, "Topology-aware virtual network embedding through bayesian network analysis," 2012.
- [14] Y. Wen, T. Lee, Z. Zhu, and L. Gong, "Toward profit-seeking virtual network embedding algorithm via global resource capacity," 2014.
- [15] M. Yu, Y. Yi, J. Rexford, and M. Chiang, "Rethinking virtual network embedding: substrate support for path splitting and migration," *Computer Communication Review*, 2008.
- [16] E. Zegura, K. L. Calvert, and S. Bhattacharjee, "How to model an internetwork," 1996.
- [17] S. Zhang, Z. Qian, J. Wu, and S. Lu, "An opportunistic resource sharing and topology-aware mapping framework for virtual networks," 2012.
- [18] S. Zhang, Z. Qian, J. Wu, S. Lu, and L. Epstein, "Virtual network embedding with opportunistic resource sharing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 3, pp. 816–827, 2014.
- [19] Y. Zhu and M. Ammar, "Algorithms for assigning substrate network resources to virtual network components," 2006.