

Received September 1, 2020, accepted September 23, 2020, date of publication September 28, 2020, date of current version October 8, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3027432

VNE-SDN Algorithms for Different Physical Network Environments

SHEN LING¹, (Member, IEEE), WU MUQING¹, (Member, IEEE),
AND XIAOLAN HOU¹, (Member, IEEE)

School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China

Corresponding author: Shen Ling (lingshen@bupt.edu.cn)

This work was supported in part by the 111 Project under Grant B17007, in part by the Beijing Key Laboratory of Network System Architecture and Convergence under Grant 2017BKL-NSAC-ZJ-01, and in part by the National Natural Science Foundation of China (NSFC) under Grant 61872401.

ABSTRACT Network virtualization (NV) is widely considered as one of the key technologies for future networks, which allows multiple virtual networks (VNs) to run on the same substrate network simultaneously. Software defined networking (SDN) can be used as a platform to realize NV. Therefore, the NV-based SDN has attracted intensive attention from both academic and industry in recent years. In this article, we focus on virtual network embedding (VNE) problem, which is one of the most important technologies of NV. Specifically, resource situation of underlying network (RSUN) is not considered in previous VNE algorithms, which seriously affects the efficiency of VNE. To solve this problem, we develop two heuristic algorithms to select the appropriate VNE according to RSUN. Our proposed algorithms adopt a novel node-ranking method, which fully considers the node topology and resource attributes, to sort physical nodes. In addition, two sorting approaches for virtual nodes and the global matching embedding (GME) approach are proposed. Our numerical analyses validate our proposed schemes and algorithms in terms of the average revenue to cost ratio, the average revenue, and the average VN request acceptance ratio.

INDEX TERMS Network virtualization, virtual network embedding, SDN, global resource matching, node-ranking approach.

I. INTRODUCTION

To overcome the shortcomings of traditional network architectures, researchers propose the software defined networking (SDN) technology, which is considered as a networking paradigm of separating the control plane from the data plane. The open and programmable interfaces allow for flexible interactions between the networking applications and the underlying physical network. As one of the enabling technologies for SDN [1], network virtualization (NV) allows multiple heterogeneous virtual networks (VNs) to run on a shared substrate networks simultaneously, which is considered as one of the key technologies for future networks. Since the virtualization of SDN networks allow networks to leverage the combined benefits of SDN and NV [2], it attracts significant research attention in recent years. Compared with the traditional network, a SDN architecture is more suitable to realize NV technologies. The main reasons are as follows:

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Khalil Afzal¹.

- 1) SDN allows third-party to develop and deploy new management modules, thus it is more convenient to deploy NV;
- 2) It can process tasks in parallel. Specifically, the centralized management mode makes parallel operation of different virtual machines easier;
- 3) In SDN, the controller plays the role of global resource control and provides a convenient condition for real-time resource acquisition. Therefore, it is significant to study NV technologies in an SDN architecture.

Fig. 1 shows a virtualization-enabled SDN architecture, which is composed of three layers. The functional descriptions of these three layers are as follows: 1) Physical network layer: Once being selected as the virtual network requests' embedding nodes or links, the corresponding substrate nodes and links, according to the forwarding rules received from the SDN controller, can be responsible for receiving user data flows and forwarding these flows; 2) Control Layer: Control layer consists of a number of SDN controllers, which are responsible for designing forwarding rules and forwarding the rules to the substrate switches. In addition, the NV hyper-visor AdVisor is deployed at the controllers to implement

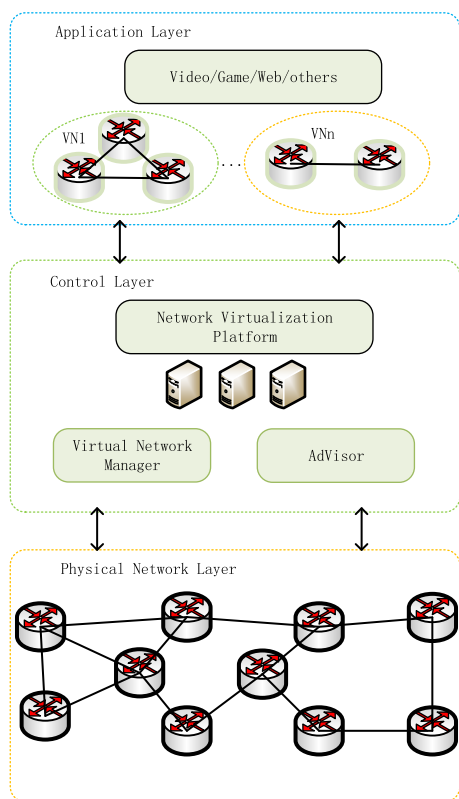


FIGURE 1. Illustration of the virtualization-enabled SDN architecture.

virtual network embedding (VNE); 3) Application Layer: Various kinds of tenant applications are handled in application layer, such as video application, game application and web application, etc. User applications are characterized as different VN requests, and embed the VNs into substrate network to implement different types of user applications.

In this paper, we mainly focus on VNE, which is one of the most critical technologies in NV. Different from the traditional network, the flow table resource needs to be extra considered in SDN network environment. Therefore, VNE problem is more complicated in SDN architectures. However, to our best of knowledge, the existing VNE algorithms have not taken into account the resource situation of underlying network (RSUN). Node resources and link resources in the underlying network cannot always remain balanced, and VNE often fails due to a certain resource shortage. Therefore, the embedding effect of previous VNE algorithms is limited due to the change of RSUN. In order to deal with this issue, we design two heuristic algorithms for physical networks with shortage of node resources (SNRs) and shortage of link resources (SLRs), and two heuristic algorithms are denoted as SNR-VNE and SLR-VNE. Bandwidth distance (BD) and resource matching degree (RMD) are defined to sort physical nodes. We first sort physical nodes according to BD, and then physical nodes with equal BD are reordered according to RMD. In SNR-VNE, the virtual nodes are sorted according to the connection of the virtual links, and the global matching embedding algorithm (GME) is proposed to embed virtual

nodes when the virtual node has no connection with the already sorted virtual nodes. In SLR-VNE, the virtual nodes are sorted according to the virtual node resources. Overall, our major contributions in this paper are as follows:

1) A novel physical node-ranking approach is proposed to sort physical nodes. We define BD and RMD in the approach. The steps are as follows: (1) Physical nodes are first sorted according to BD; (2) According to RMD, we rearrange the order of nodes with the equal BD.

2) We design a novel node-embedding approach, namely the GME algorithm, which can save CPU resources to the greatest extent. We calculate the RMD between all physical nodes and virtual nodes, and the most matching virtual node and physical node are preferentially paired.

3) Two sub-algorithms are proposed to solve VNE for different RSUNs. These two algorithms are applied to different RSUNs, which are not considered in previous VNE algorithms.

4) We set up two network environments to validate the advantage of our proposed algorithms. Simulation results show that our proposed algorithms outperform other algorithms in terms of average revenue to cost (R/C) ratio, average long-term average revenue, and average acceptance ratio, respectively.

The rest of the paper is organized as follows. Section II briefly describes the related work. Section III presents the network model and formulates the problem of the VNE. Sections IV presents the integer programming formulation to solve VNE problems. Section V introduces the node sorting algorithm and global matching matrix (GMM). Section VI introduces our proposed SNR-VNE and SLR-VNE algorithms. Section VII provides performance evaluations. The paper concludes with Section VIII.

II. RELATED WORK

In this section, some works closely related to our work are introduced. We first discuss the VNE heuristic algorithms based on the classical method of node ordering, and then discuss the VNE heuristic algorithms in a SDN architecture.

A. THE CLASSICAL METHOD OF NODE RANKING

In [3], the Markov Random Walk (RW) model is applied to rank a network node based on its resource and topological attributes. The node ranking measure reflects the relative importance of physical nodes. According to the node ranking, the authors propose two VNE algorithms, namely RW-Max Match and RW-BFS. RW-Max Match embeds virtual nodes onto substrate nodes according to the above ranks, and then embeds the virtual links by shortest path method. RW-BFS is a backtracking VNE algorithm based on breadth-first search, which embeds virtual nodes and links during the same stage by using node-ranking.

In [4], the authors propose a VNE algorithm named VNE-NTANRC, which considers both the network topology attributes and network resources. In VNE-NTANRC, a novel node-ranking approach is adopted to rank all substrate

and virtual nodes before embedding each VN. The novel node-ranking approach consists two sub-approaches and considers five important network topology attributes and global network resources altogether. One sub-approach calculates all node values (NoVs) directly according to the node-ranking approach. The other sub-approach is stimulated through the Google PageRank website algorithm, and can calculate NoV in a stable state. In [5], network centrality analysis is introduced into the VNE, and VNE algorithms are proposed based on closeness centrality. The algorithms are more reasonable than previous works in coordinating node and link embedding. In addition, the proposed VNE approaches improve the network utilization efficiency largely and decrease the time complexity.

B. APPLICATION OF ARTIFICIAL INTELLIGENCE TECHNOLOGY IN VNE

In recent years, with the development of artificial intelligence, scholars introduce reinforcement learning into the problem of VNE to optimize the node mapping process.

In [6], authors propose an efficient online heuristic VNE algorithm, named Presto, based on an artificial intelligence resource abstraction model called Blocking Island. This is the first to apply Blocking Island paradigm to solve the VNE problem. In [7], authors design reinforcement learning-based neuro-fuzzy algorithms that perform dynamic, decentralized and coordinated self-management of substrate network resources to achieve better efficiency in the utilization of substrate network resources. In addition, the proposed algorithms are evaluated through comparisons with a Q-learning-based approach. A policy network based on reinforcement learning is designed and implemented in [8]. Authors use policy gradient to achieve optimization automatically by training the policy network with the historical data based on virtual network requests. A Continuous-Decision VNE scheme relying on Reinforcement Learning (CDRL) is proposed in [9], which regards the node embedding of the same request as a time-series problem formulated by the classic seq2seq model. Moreover, two traditional heuristic embedding algorithms as well as the classic reinforcement learning aided embedding algorithm are used for benchmarking the CDRL algorithm. In [10], authors propose a security-aware VNE algorithm based on reinforcement learning. Security requirement level constraint is added for each virtual node, and each substrate node. Virtual nodes are embedded on substrate nodes that are not lower than the level of security requirements. In [11], authors propose a new and efficient algorithm, which combines deep reinforcement learning with a novel neural network structure. In [12] reinforcement learning based prediction model is designed for the efficient Multi-stage Virtual Network Embedding (MUVINE) among the cloud data centers.

C. THE VNE HEURISTIC ALGORITHMS IN SDN ENVIRONMENT

Hypervisors for SDN networks are comprehensively surveyed in [2]. The SDN hypervisors first are categorized

according to their architectures into centralized and distributed hypervisors, and then the hypervisors are further sub-classified according to their execution platform into hypervisors running exclusively on general-purpose compute platforms, or on a combination of general-purpose compute platforms with general- or special-purpose network elements. In addition, the authors exhaustively compare the network attribute abstraction and isolation features of the existing SDN hypervisors. Finally, the authors outline the development of a performance evaluation framework for SDN hypervisors.

A hierarchical virtualization-enabled SDN architecture is proposed in [1]. Under the constraints of virtual network requirements and the resource characteristics of substrate network, the VNE is formulated as a multi-objective optimization problem to minimize network load and maximize embedding reliability. Since optimization problem is a multi-objective optimization problem, which is relatively complicated, the ideal point method is applied to solve this problem. Specifically, virtual node embedding sub-algorithm is first proposed to determine the locally optimal solution to the two sub-problems, namely network load minimization and embedding reliability maximization sub-problem. Then, according to the distance between the feasible solutions and the locally optimal solutions, a single-objective optimization problem is formulated. To obtain the global VNE strategy, a discrete particle swarm optimization (DPSO) algorithm is applied to solve the problem. A self-adaptive VNE algorithm is proposed in [13]. The authors divide VN requests into different types through the adaptive algorithm. The authors propose a self-adaptive VNE algorithm, through which VN requests are divided into different types, and an integer linear programming is formulated to solve VNE problem. In addition, three different types of VN requests are considered, i.e., VN requests for high bandwidth requirements, VN requests for low latency requirements, VN requests for high bandwidth requirements and latency requirements. In [14], the authors tackle two problems, which are VNE to balance the load on the substrate network and controller placement to minimize controller-to-switch delays. In [15], authors propose a mathematical programming formulation that considers multi-objectives. In addition, a heuristic algorithm is developed by proposing new design metrics.

D. BRIEF SUMMARY

To summarize, VNE has achieved some results in traditional network architecture. Since a VNE problem is NP-hard in the traditional network environment, heuristic algorithms are developed to solve VNE problems. To improve embedding efficiency, authors propose some efficient node-ranking approaches before embedding virtual nodes. However, only node capacity and link bandwidth requirements are considered in traditional network environments. Comparing with the traditional network environment, the SDN environment needs to consider the constraint flow table attribute. Thus, the VNE problem is much more complicated than traditional

network environment of VNE. RSUN is not considered in the existing algorithms, thus algorithms are not suitable for SDN environment. To deal with this issue, we propose two VNE algorithms, namely SNR-VNE and SLR-VNE for different RSUNs.

In this paper, our proposed SNR-VNE and SLR-VNE algorithms differ from previous algorithms in three aspects. First of all, SNR-VNE and SLR-VNE are designed for scarce node resources and scarce bandwidth resources of physical networks, respectively. Virtual network requests from different tenants, which have different requirements on node resources and link resources, may cause uneven network resources. Thus, it is significant to develop VNE algorithms for different underlying network environments. To the best of our knowledge, this is the first time to propose the VNE algorithms for different network resource conditions. Second, our proposed algorithms adopt a novel node-ranking method, namely Best-Match approach to rank substrate nodes before embedding each virtual node. The novel node-ranking approach takes two parts named Bandwidth-Distance and Resource-Variance into consideration. Physical nodes are first sorted according to the value of Bandwidth-Distance, and then the order of physical nodes, which have equal Bandwidth-Distance values are updated according to the value of Resource-Variance. This node-ranking approach increases the possibility of embedding the virtual nodes onto substrate nodes that are close to each other, and selects substrate node that has a higher matching degree with virtual node. Finally, the difference between SNR-VNE and SLR-VNE is reflected in the embedding order of virtual nodes. As to the SNR-VNE algorithm, we embed the virtual nodes with larger node resource requirements, which are relatively difficult to be embedded. Due to the relative shortage of node resources, the Global-Matching algorithm is proposed to embed each virtual node. The node embedding algorithm can save node resources to the greatest extent. As to the SLR-VNE algorithm, virtual nodes are ranked according to link bandwidth.

III. NETWORK MODEL AND PERFORMANCE METRICS

In this section, the substrate network and virtual network request models are described firstly, and then the measurements of network resources are introduced in the following subsection. The main index notations in this paper are listed in Table 1.

A. NETWORK MODEL

To describe the VNE problem in the SDN environment, we model the substrate network as a weighted undirected graph and denote it by $G_s = (N_s, L_s, R_s^n, R_s^l)$, where N_s denotes the set of substrate nodes, L_s denotes the set of substrate links, R_s^n denotes the set of attributes in the substrate node $n_s \in N_s$ including the CPU capacity $CPU(n_s)$ and flow table capacity $F(n_s)$, and R_s^l denotes the attributes of the substrate links $l_s (l_s \in L_s)$. We consider the link bandwidth $BW(l_s)$ and delay $Delay(l_s)$ as the link attributes. The lower part of Fig. 2 represents a substrate network. The total CPU

TABLE 1.

Notation	Description
c	The Substrate Network
N_s	100
n_s, i, j	The Substrate Nodes
L_s	The Set of Substrate Links
l_s, ij	The Substrate Links
R_s^n	The Set of Substrate Node Attributes
R_s^l	The Set of Substrate Node Attributes
$CPU(n_s)$	The Total CPU of Substrate Node n_s
$F(n_s)$	The Total Flow Rules of Substrate Node n_s
$BW(l_s)$	Total Bandwidth of Substrate Link l_s
$Cpu(n_s)$	The Available CPU of Substrate Node n_s
$Flow(n_s)$	The Available Flow Rules of Substrate Node n_s
$Bw(l_s)$	The Available Bandwidth of Substrate Link l_s
$Delay(l_s)$	The Link Propagation Delay of the Substrate Link l_s
P_{AE}	The Set of Paths that Start from Source Node A to End Node E
G_v	The Virtual Network
N_v	The Set of Virtual Nodes
N_v, u, v	The Virtual Nodes
L_v	The Set of Virtual Links
l_v, uv	The Virtual Links
R_v^n	The Set of Virtual Node Attributes
R_v^l	The Set of Virtual Link Attributes
$Cpu(n_v)$	The CPU Requirement of Virtual Link n_v
$Flow(n_v)$	The Flow Rules Requirement of Virtual Node n_v
$Bw(l_v)$	The Bandwidth Requirement of Virtual Link l_v
$Delay(l_v)$	The Link Propagation Delay Requirement of the Virtual Link l_v
M_G^s	The Set of Physical Nodes have Embedded the Virtual Nodes Which belong to VN G
M_G^v	The Set of Virtual Nodes have Embedded the Physical Nodes Which belong to VN G

and flow table are indicated on the left part of the box, and the residual CPU and flow rules are represented on the right part of the box. The numbers over the links denote the total bandwidth resources, whose right side number represents the residual bandwidth resources.

Similar to the substrate network, the virtual network can be modeled as a weighted graph $G_v = (N_v, L_v, R_v^n, R_v^l)$, where N_v is the set of all virtual nodes and L_v is the set of all virtual links, R_v^n denotes the set of attribute in the virtual node $n_v \in N_v$ including the required computing capacity $CPU(n_s)$ and flow table capacity $F(n_s)$, R_v^l denotes the required bandwidth capacity. We consider the required bandwidth $BW(l_s)$ and delay $Delay(l_s)$ as the link attributes. The top part of Fig. 2 represents two virtual networks. The numbers aside the nodes represent the required CPU and flow rules, respectively. The numbers over the links represent the required bandwidth resources.

We use $M(v) = n$ to represent that the virtual node v is embedded into the physical node n . The embedding process is typically decomposed into two embedding steps as follows: (1) embedding the virtual node to the physical node that can satisfy the virtual node resource constraints; (2) embedding the virtual link to the substrate links that can satisfy the virtual link resource constraints. It is worth noting that the virtual

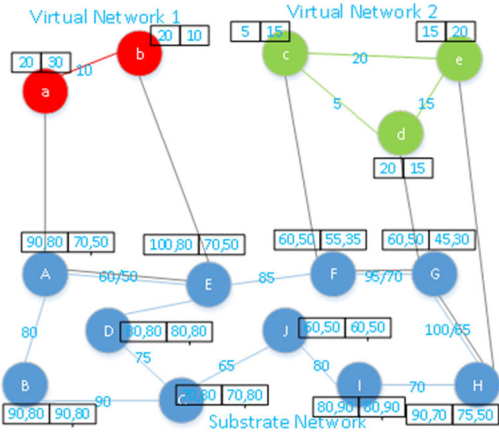


FIGURE 2. An example of VN embedding.

nodes from different virtual network requests can be embedded to the same substrate nodes, but cannot be embedded on the same substrate nodes when the virtual nodes come from the same virtual network request. In fact, these two stages are not sequential when we solve the problem of VNE. Fig. 2 indicates the VNE solutions. Take VN1 as an example, the node embedding solution is $\{a \rightarrow A, b \rightarrow B\}$, and link embedding solution is $\{l_{ab} \rightarrow p_{AE}\}$.

B. PERFORMANCE METRICS

In order to evaluate the performance of VNE algorithms, performance metrics are necessary to be defined. This subsection introduces some important performance metrics commonly used in VNE. Similar to previous works [16], the revenue of a VN request is defined as follows:

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

Accordingly, the cost of infrastructure providers (InPs) accommodating a VN request can be defined as follow:

$$C(G_v) = \sum_{n_v \in N_v} \{CPU(n_v) + H(n_v)\} + \sum_{l_v \in L_v} BW(l_v) \times hops(l_v) \quad (2)$$

From the InP’s perspective, the main goal of online VN embedding algorithm is to maximize the revenue of InPs and increase the resource utilization of substrate network by accommodating as more VN requests as possible or improving the R/C ratio in the long run. We define three performance metrics to evaluate our proposed algorithms, including the long-term average revenue, R/C ratio and the VN request acceptance ratio.

The long-term average revenue can be formulated as follows:

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

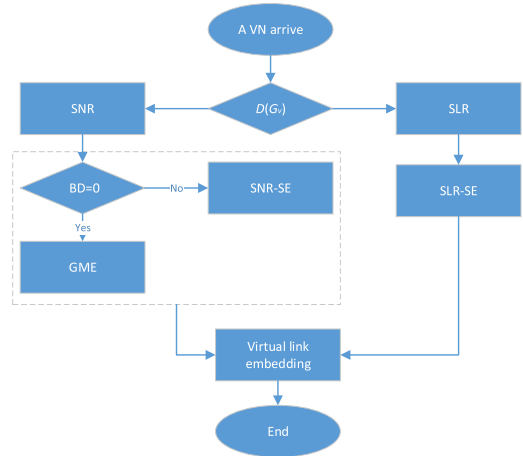


FIGURE 3. The diagram of VNE process.

The operator’s profit can be expressed by:

$$R/C = \frac{\sum_{t=0}^T R(G_v, t)}{\sum_{t=0}^T C(G_v, t)} \quad (4)$$

The VN request acceptance ratio can be defined by:

$$acce = \lim_{t \rightarrow \infty} \frac{\sum_{t=0}^T VN_{accept}}{\sum_{t=0}^T VN_{request}} \quad (5)$$

where VN_{accept} denotes the number of already accepted VN requests, and $VN_{request}$ denotes the number of arrived VN requests.

IV. PROPOSED VNE ALGORITHM

The proposed VNE algorithm is detailed in this section. The overall framework is introduced at first, then we present methods of sorting physical nodes and virtual nodes. Next, we present GME. Finally, two heuristic algorithms are designed for different network resource conditions.

A. THE FRAMEWORK OF PROPOSED VNE ALGORITHM

The proposed VNE framework is presented in Fig. 3. When a VN arrives, the system first determines the type of RSUN. Equation (6) defines $D(G_v)$, which denotes RSUN between VN and physical network. We select SNR-VNE as the embedding method, when $D(G_v) \leq \theta$; otherwise, SLR-VNE is selected as embedding method. θ is RSUN’s threshold, and it is a real number. From the Fig. 3, the difference between the two sub-algorithms is at the node embedding stage. The node embedding phase of SNR-VNE contains GME and sorting embedding algorithm. In SNR-VNE, the sorting embedding algorithm is denoted as SNR-SE. In SLR-VNE, the sorting

Algorithm 1 Physical Node Ranking Algorithm

Input: G_s, G_v .
Output: Node-Ranking vector τ of the given network G

1. **For** all physical nodes i
2. Calculate BD;
3. **End for**
4. According to the value of BD, sort the substrate nodes in descending order and denote it by τ ;
5. Calculate the RMD for the nodes which have the equal BD value;
6. According to the value of RMD, sort the substrate nodes which have the equal BD value, in descending order, and update τ .

embedding algorithm is denoted as SLR-SE.

$$D(G_s) = \frac{\sum_{n_s \in G_s} (CPU(n_s) + Flow(n_s))}{\sum_{l_s \in G_s} Bw(l_s)} \quad (6)$$

B. NODE RANKING APPROACH

1) PHYSICAL NODE RANKING APPROACH

In order to embed virtual node u , the physical nodes are sorted according to BD. Formula (7) defines the BD of physical node i .

$$BD_i = \sum_{v \in N_v} distance(i, j) * BW(l_{uv}), M(v) = j \quad (7)$$

where $distance(i, j)$ represents the hops of the shortest path, $BW(l_{uv})$ denotes the connection bandwidth of virtual link l_{uv} , j denotes the physical node, which host virtual node v , and v denotes any other node, which belongs to the same virtual network as u .

In an SDN environment, flow rules need to be considered. In order to avoid the unbalanced consumption of CPU and flow rules, $RBR_{i,v}$ and $Var_{i,v}$ are defined. $RBR_{i,v}$ denotes the distance between the resources of physical node i and virtual node v . $Var_{i,v}$ reflects the distance between the resources of physical node i and virtual node v . The values of $Var_{i,v}$ and $RBR_{i,v}$ can be obtained according to Equation (8) and (9). We can get $RMD_{i,v}$ according to Equation (10).

$$RBR_{i,v} = \frac{\min[F(i)/CPU(i), F(v)/CPU(v)]}{\max[F(i)/CPU(i), F(v)/CPU(v)]} \quad (8)$$

$$Var_{i,v} = [F(i) - F(v)]^2 + [CPU(i) - CPU(v)]^2 \quad (9)$$

$$RMD_{i,v} = \frac{RBR_{i,v}^2}{\sqrt{Var_{i,v}}} \quad (10)$$

According to the value of BD, physical nodes first are sorted in ascending order. However, there are many physical nodes that have equal values of BD. Therefore, we calculate the RMD for the nodes with equal BD value, and sort the nodes in descending order according to the value of RMD. The procedures of the physical node-ranking approach are detailed in Algorithm 1.

Algorithm 2 Virtual Node Ranking in SLR-VNE

Input: G_s, G_v .
Output: NS link embedding solution.

1. **For** all virtual nodes do
2. Calculate $H_2(n)$;
3. **End for**
4. The node with the highest $H_2(n)$ as the first embedding node, recorded as $A(1)$;
5. **For** $n = 2: |N_v|$
6. **For** $m = 1: |N_v| - n$
7. Calculate H'_2 ;
8. **End for**
9. The node with the highest H'_2 as the priority embedding node, recorded as $A(n)$;
10. **End for**

2) VIRTUAL NODE RANKING APPROACH

We propose two virtual node ranking approaches for different underlying networks. The methods are applied to the physical network where the node and link resources are scarce. We need to save as many node resources as possible for SNR network. For the physical network with scarce node resources, the embedding algorithm should preferentially embed virtual nodes with large resource requirements. We sort the virtual nodes according to Equation 11.

$$H(n_v) = CPU(n_v) + F(n_v) \quad (11)$$

In SLR-VNE, we first embed the virtual node with the largest connection bandwidth, which can be obtained through Formula (12). The embedding order of the remaining virtual nodes is determined according to Formula (13). Virtual nodes are sorted according to algorithm 2. A (Line 4 and Line 9) represents the set of nodes that have been sorted.

$$H_2(n) = \sum_{l_v \in nbr(n)} BW(l_v) \quad (12)$$

$$H'_2(n) = \sum_{m \in A} BW(l_{mm}) \quad (13)$$

C. GME ALGORITHM

The case of $BD_{i,v} = 0$ should be considered as to SNR-VNE. In addition to the first virtual node, other virtual nodes may have no virtual link that connects with the embedded virtual nodes. When $BD_{i,v}$ is equal to 0, it results in inefficient embedding because physical nodes cannot be sorted. SNR-VNE adopts the global matching embedding (GME) algorithm to embed the virtual node. GME algorithm is detailed in Algorithm 3.

D. HEURISTIC ALGORITHM

1) SNR-VNE ALGORITHM

According to the above node ranking, two sub-algorithms are designed for different network resource conditions. Our proposed algorithms include two-stage, namely node embedding stage and the link embedding stage. SNR-VNE is a

Algorithm 3 GME Algorithm

Input: G_s, G_v .
Output: Global optimal solution.

1. **For** $v = 1: |N_v|$
2. **For** $i = 1: |N_s|$
3. Calculate $RMD_{i,v}$ base on Eq (10);
4. **End for**
5. **End for**
6. Obtaining global matching matrix
7. Sort matrix elements in descending order, and record the corresponding virtual nodes and physical nodes, recorded as Q;
8. **For** $i = 1: |N_s||N_v|$
9. Select the top GMM, and find the corresponding virtual node and physical node;
10. **If** $CPU(N_v) \leq CPU(N_s)$ and $F(N_v) \leq F(N_s)$;
11. $M(n_v) = n_s$;
12. Delete the virtual node n_v in τ , and set the rows, which correspond to the embedding virtual nodes in GMM to 0;
12. **Break**;
13. **Else**
14. Delete the top Q, and set the corresponding element in GMM to 0;
15. **End if**
16. **End for**

Algorithm 4 SNR-VNE Algorithm

Input: G_s, G_v .
Output: NS node embedding solution.

1. According to Formula (11), virtual nodes are sorted in descending order and denote it by τ ;
2. Calculate the GMM, according to Algorithm 2;
3. **For** $v = 1: |N_v|$
4. Select the top virtual node n_v in τ ;
5. **If** the maximum BD value is equal to 0
6. Use GME to embed virtual nodes;
7. **If** the virtual node i is successfully embedded
8. Delete virtual node i in τ ;
9. **End if**
10. **Else**
11. Sort physical nodes according to algorithm 1, and denote it by Ω ;
12. **For** $j = 1: |N_s|$
13. **If** $CPU(n_v(i)) \leq CPU(n_s(j))$ and $F(n_v(i)) \leq F(n_s(j))$;
14. $M(n_v) = n_s$;
15. **Break**;
16. **End if**
17. **End for**
18. **End if**
19. **End for**

two-stage VN embedding algorithm. Algorithm 4 presents the node embedding procedure of SNR-VNE scheme, which

Algorithm 5 LBM-VNE Link Embedding Algorithm

Input: G_s, G_v .
Output: NS link embedding solution.

1. Sort the virtual links by the required bandwidth in non-increasing order;
2. **For** all the unembedded virtual links in VN do
3. Choose the virtual link L_v with the highest $BW(L_s)$;
4. Use Dijkstra shortest path algorithm to find the shortest path;
5. **End for**

Algorithm 6 SLR-VNE Node Embedding Algorithm

Input: G_s, G_v .
Output: LS node embedding solution.

1. Sort virtual nodes according to algorithm 2;
2. **For** $i = 1: |N_v|$
3. Sort physical nodes according to algorithm 1, and denote it by Ω ;
4. **For** $j = 1: |N_s|$
5. **If** $CPU(n_v(i)) \leq CPU(n_s(j))$ and $F(n_v(i)) \leq F(n_s(j))$
6. $M(n_v) = n_s$;
7. **Break**
8. **End if**
9. **End for**
10. **End for**

contains two parts. When the maximum BD value is equal to 0, we adopt the GME algorithm to embed virtual nodes; otherwise, we adopt the ranking method to embed the virtual nodes according to Equation 11, and physical nodes are sorted according to Algorithm 4 (Lines 11-17).

In the link embedding phase, we prioritize embedding virtual links with greater bandwidth requirements. SNR-VNE performs the link embedding by using shortest path algorithm to embed the virtual links to the substrate paths. Virtual link embedding algorithm is detailed in Algorithm 5.

2) SLR-VNE ALGORITHM

SLR-VNE is also a two-step embedding algorithm. We first sort the virtual nodes according to Algorithm 2, and then sort the substrate nodes according to Algorithm 1. Finally, the qualified physical node is selected as the embedding node. The detailed steps are shown in Algorithm 6.

Like the link embedding in the SNR-VNE algorithm, the shortest path algorithm is used to embedding virtual links in SLR-VNE.

E. TIME COMPLEXITY ANALYSIS

In order to reduce the complexity of the algorithm, we calculate the distance of all nodes in advance. For the SNR-VNE, the total time complexity involves three main procedures, namely the physical node ranking, GME and link embedding. Therefore, the total time complexity can be represented as $O(|N_s||N_v|^2 + |N_s|^2|L_v|)$. For the SLR-VNE, it consists of the three main steps, namely the physical node ranking, virtual

TABLE 2.

Physical Network	Parameters
Experiment 1: Network Generation Approach	Salam method, BorderLength=1000, Alpha=10, Beta=19
Experiment 2: Network Generation Approach	Salam method, BorderLength=1000, Alpha=10, Beta=0.35
Node Capacity	[50,100], uniform distributed
Link Bandwidth	[50,100], uniform distributed

TABLE 3.

Virtual Network	Parameters
Network Generation Approach	Salam method, Border Length=1000, Alpha=10, Beta=19
VNR Arrival Rate	4 VNRs per 100 time units
Number of virtual nodes	An integer, distributed[2,20]
Node Capacity Demand	[1,50], uniform distributed
Link Bandwidth Demand	[1,50], uniform distributed

node ranking, and link embedding. Therefore, the total time complexity can be represented as $O(|N_s||N_v| + |N_s|^2|L_v|)$.

V. PERFORMANCE EVALUATION

This section presents the simulation parameter settings in the form of tables. Main simulation results are illustrated. In addition, we elaborate on quantifying the advantage of the SNR-VNE and SLR-VNE algorithm. Our simulation work composes of two parts. The aims of first part is to prove that the embedding efficiency of SNR-VNE is greatly improved compared with other algorithms. This part use substrate network where each pair of substrate nodes is connected with the probability of 0.5. The other part is to use substrate network with 500 links, which elaborates on highlighting the efficiency and effectiveness of SLR-VNE.

A. SIMULATION PARAMETER SETTINGS

To evaluate SNR-VNE and SLR-VNE, we conduct the simulation work in a self-developed platform. Table 2 summarizes the shared substrate network (single domain) parameters for our simulation. These parameters are typical in VNE research area. Table 3 summarizes the VN parameters for our simulation. In this paper, we set the value of θ to 0.1.

B. COMPARED ALGORITHMS

Four heuristic VNE algorithms make up the simulation part in total. NRM-VNE and RCR-VNE are proposed in [16]. These algorithms are typical, latest, and most-related to our algorithms.

C. SIMULATION RESULTS

According to $D(G_v)$, the simulation experiment 1 is based on physical network practices where node resources are relatively scarce. In this section, main simulation results are presented. Fig. 4 shows the average R/C ratio as a function of time, Fig. 5 shows the average revenue as a function of time, and Fig. 6 presents the average VN acceptance ratio

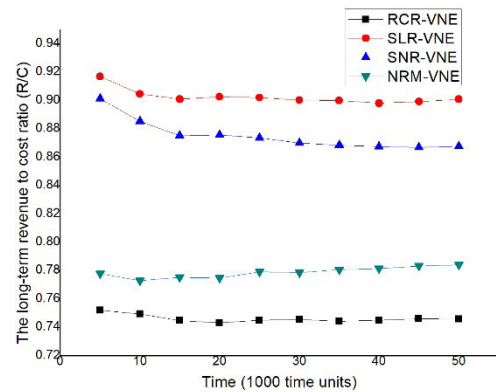


FIGURE 4. Average R/C ratio.

as a function of time. These three figures aim to prove the efficiency and effectiveness of SNR-VNE in the long term.

1) AVERAGE R/C RATIO

Fig. 4 plots the average R/C ratio in terms of time. Derived from Fig. 4, the average R/C ratio of all selected algorithms decreases with the variation on time. As for RCR-VNE, the R/C ratio is lowest, because adjacent virtual nodes are embedded onto substrate nodes that are far from each other. Thus, it leads to large amount of unnecessary bandwidth resources consumption. As to the NRM-VNE algorithm, considering the connection bandwidth of the nodes, bandwidth resources can be saved. Compared with NRM-VNE and RCR-VNE algorithms, SNR-VNE and SLR-VNE perform better. It is owing to that the distance between the nodes is considered in our proposed algorithm. For the SLR-VNE, the R/C ratio is the highest, because the link bandwidth of the virtual network is considered. Thus, more extra bandwidth resources are saved. Fig. 4 shows that the R/C of SNR-VNE algorithm is about 12% and 15% higher than the R/C of NRM-VNE and RCR-VNE, respectively.

2) AVERAGE REVENUE

Fig. 5 depicts the average revenue as a function of time. To all selected algorithms, the average revenue tends to decrease with the number of VNs increasing. The SNR-VNE has an apparent advantage over other heuristic algorithms. The reason that the SNR-VNE algorithm has a larger average benefit than other heuristic algorithms is that SNR-VNE can accept more VNs than other heuristic algorithms. Because SNR-VNE considers local resource matching and global resource matching in the node embedding phase, which helps to increase the utilization of node resources. When the number of VNs increases, the SNR-VNE is able to embed VNs more effectively. However, the performance of SLR-VNE is very general. Because in the node embedding phase, SLR-VNE does not consider the global resource matching, but consider the connection bandwidth of the virtual network. Although SLR-VNE performs best at the R/C ratio, the performance of SNR-VNE is more stable when the three

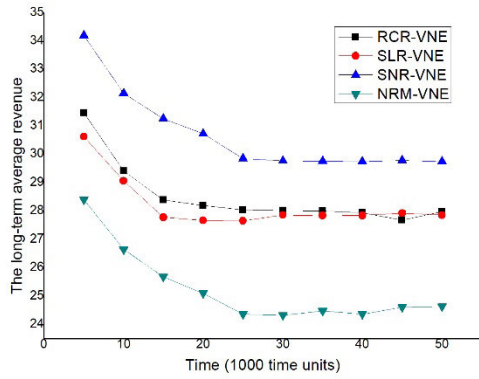


FIGURE 5. Average revenue.

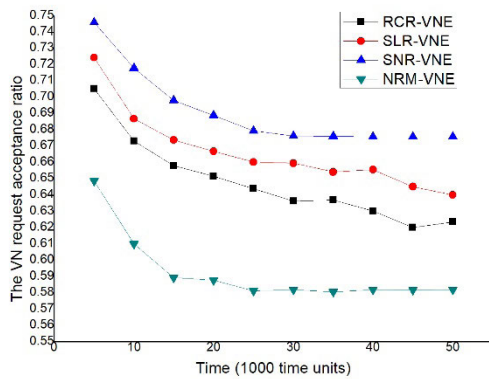


FIGURE 6. Average VN acceptance ratio.

performance metrics are considered together. Therefore, SNR-VNE is more suitable for the physical network with insufficient node resources. Fig. 5 shows that SNR-VNE generates 5% and 10% higher revenue than RCR-VNE and NRM-VNE, respectively.

3) AVERAGE VN ACCEPTANCE RATIO

Fig. 6 plots the average VN acceptance ratio as a function of continuous time. The acceptance ratio is an important metric to evaluate different VNE algorithms' embedding abilities in a continuous time event. As shown in Fig. 6, the average acceptance ratio of all algorithms almost decay with the variation on time. It reveals that there are no infinite node resources for embedding more VNs. In addition, our proposed SNR-VNE algorithm outperforms all selected heuristic algorithms. It runs as expected because the SNR-VNE takes local resources matching and global resources matching into account simultaneously. Through using local resources matching approach, virtual nodes are embedded to physical nodes with high resource matching degrees when the topological distances of physical nodes are the same; and through using global resources matching approach, the best matching virtual nodes and physical nodes are found and embedded. Since the physical node resources are relatively scarce, it is beneficial to use node resources efficiently. An efficient node embedding is therefore likely to be achieved.

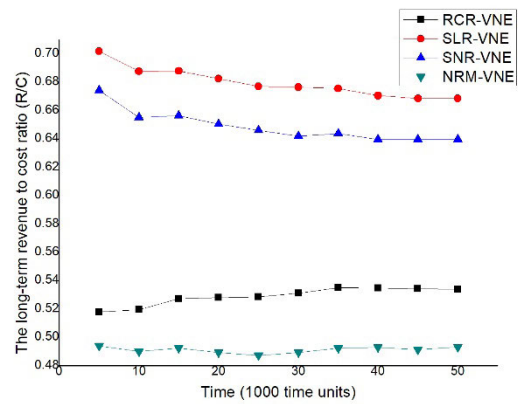


FIGURE 7. Average R/C ratio.

However, the SLR-VNE algorithm ranks virtual nodes according to the connection bandwidth of the virtual network. In the case of node resources shortage, connection bandwidth sort cannot efficiently use node resources. Therefore, SLR-VNE is not able to perform as well as the SNR-VNE algorithm. To the remaining heuristic algorithms (NRM-VNE and RCR-VNE), only physical node resource attribute is considered in the node embedding stage. Therefore, they cannot perform well in physical networks where node resources are scarce. Fig. 6 shows that the acceptance ratio of SNR-VNE is about 5% and 10% higher than the acceptance ratio of RCR-VNE and NRM-VNE, respectively.

According to $D(G_v)$, simulation experiment 2 is executed on the physical network with relatively scarce bandwidth resources. Similar to experiment 1, we select the R/C ratio, average revenue and VNs acceptance ratio to discuss the efficiency and effectiveness of SLR-VNE in the long term.

4) AVERAGE R/C RATIO

Fig. 7 plots the R/C ratio as a function of VNs arrival rate. The R/C ratio metric is used to evaluate the resource efficiency of different embedding algorithms. For SNR-VNE and SLR-VNE, the average R/C ratio tends to decrease at the beginning of the embedding. Physical resources are fragmented when our algorithms perform VNE in the early time of the simulation, and this leads to adjacent virtual nodes being embedded onto substrate nodes that are far from each other. The VN R/C ratio of all algorithms decrease to stable statuses since 30000 time units. With accommodating more VNs, a balance can be achieved. Compared with the NRM-VNE and RCR-VNE algorithms, our algorithms are able to perform better. It is owing to that the distance of physical nodes is considered in SLR-VNE and SNR-VNE algorithms. For SLR-VNE, the reason for highest R/C ratio is that the algorithm considers the link bandwidth of the virtual links. Fig. 7 shows that the R/C of SLR-VNE algorithm is about 15% and 20% higher than the R/C of RCR-VNE and NRM-VNE, respectively.

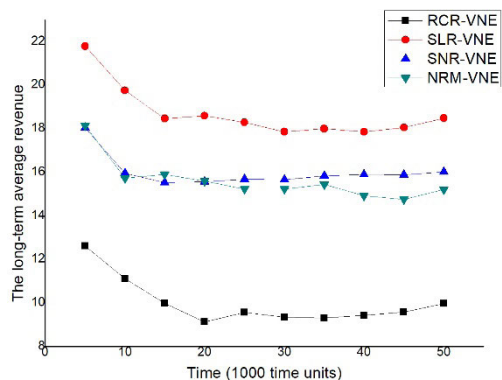


FIGURE 8. Average revenue.

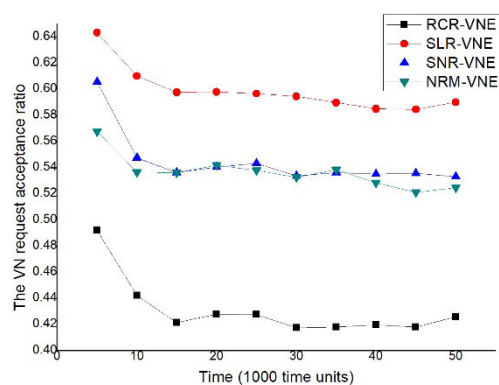


FIGURE 9. Average VN acceptance ratio.

5) AVERAGE REVENUE

We plot the revenue results of all four embedding algorithms in Fig. 8. The average revenue is essential to evaluate different VNE algorithms abilities to embed VNs. From the Fig. 8, the average revenue of all algorithms almost decay with the variation on time. For all heuristic algorithms, the average revenue will decrease to stable statuses since 20000 time units. SLR-VNE algorithm earns the highest average revenues among all the algorithms. This behavior lies in two main reasons as follows: 1) In the virtual node ranking stage, we fully consider the connection bandwidth of virtual node. We first embed virtual nodes with the largest connection bandwidth, and the order of the other virtual nodes is determined by the virtual link bandwidth; 2) In the physical node ranking phase, the underlying network's BD is considered and it embeds adjacent virtual nodes to the closer substrate nodes, which helps to save bandwidth resources and increase the acceptance rate of VNs in the case of a shortage of bandwidth resources. Fig. 8 shows that SLR-VNE generates 4% and 10% higher revenue than NRM-VNE and RCR-VNE, respectively.

6) AVERAGE VN ACCEPTANCE RATIO

Fig. 9 plots the average VN acceptance ratio as a function of continuous time. From the Fig. 9, the average VN acceptance ratio of all algorithms almost decay with the simulation time increasing. It shows that there are no infinite

bandwidth resources to accept more and more VNs, which are applied by different end-users. In addition, our proposed SLR-VNE algorithm outperforms other heuristic algorithms. It runs as expected because the SLR-VNE takes the bandwidth of virtual link and node distance of physical network into account simultaneously. For SLR-VNE, virtual nodes are embedded to physical nodes that are closer together than other heuristic algorithms. This is beneficial to improve virtual network acceptance in the physical network where bandwidth resources are scarce. However, SNR-VNE algorithm considers node resource matching and virtual node resources, but not virtual link connection bandwidth in the virtual node sorting phase. Therefore, the behavior of SNR-VNE is not as good as the SLR-VNE algorithm. Fig. 9 shows that the acceptance ratio of SLR-VNE is about 5% and 18% higher than the R/C of NRM-VNE and RCR-VNE, respectively.

VI. CONCLUSION

In this paper, we have proposed two heuristic algorithms for the two underlying network environments. In order to improve the node resource utilization of the node resource scarce network, we prefer to select a physical node with high matching as an embedding node. In addition, we use the global matching method to embed virtual nodes that are not connected to the embedded virtual nodes. For the bandwidth resource scarce network, we sort the virtual nodes according to the connection relationship and sort the physical nodes based on the number of hops. These strategies embed the adjacent virtual nodes to substrate nodes that are close to each other to increase the efficiency of bandwidth usage. The simulation results show that the VNE algorithms proposed in this paper perform better average acceptance ratio, average R/C ratio, and average revenue than the other algorithms in different network environments.

REFERENCES

- [1] R. Chai, D. Xie, L. Luo, and Q. Chen, "Multi-objective optimization-based virtual network embedding algorithm for software-defined networking," *IEEE Trans. Netw. Service Manage.*, vol. 17, no. 1, pp. 532–546, Mar. 2020.
- [2] A. Blenk, A. Basta, M. Reisslein, and W. Kellerer, "Survey on network virtualization hypervisors for software defined networking," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 655–685, 1st Quart., 2016.
- [3] 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 Comput. Commun. Rev.*, vol. 41, no. 2, pp. 38–47, Apr. 2011.
- [4] H. Cao, L. Yang, and H. Zhu, "Novel node-ranking approach and multiple topology attributes-based embedding algorithm for single-domain virtual network embedding," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 108–120, Feb. 2018.
- [5] Z. Wang, Y. Han, T. Lin, H. Tang, and S. Ci, "Virtual network embedding by exploiting topological information," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2012, pp. 2603–2608.
- [6] T. Wang and M. Hamdi, "Presto: Towards efficient online virtual network embedding in virtualized cloud data centers," *Comput. Netw.*, vol. 106, pp. 196–208, Sep. 2016.
- [7] R. Mijumbi, J. L. Gorricho, J. Serrat, M. Shen, K. Xu, K. Yang, "A neuro-fuzzy approach to self-management of virtual network resources," *Expert Syst. Appl.*, vol. 42, no. 3, pp. 1376–1390, 2015.
- [8] H. Yao, X. Chen, M. Li, P. Zhang, and L. Wang, "A novel reinforcement learning algorithm for virtual network embedding," *Neurocomputing*, vol. 284, pp. 1–9, Apr. 2018.

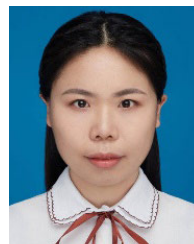
- [9] H. Yao, S. Ma, J. Wang, P. Zhang, C. Jiang, and S. Guo, "A continuous-decision virtual network embedding scheme relying on reinforcement learning," *IEEE Trans. Netw. Service Manage.*, vol. 17, no. 2, pp. 864–875, Jun. 2020.
- [10] P. Zhang, C. Wang, C. Jiang, and A. Benslimane, "Security-aware virtual network embedding algorithm based on reinforcement learning," *IEEE Trans. Netw. Sci. Eng.*, early access, May 19, 2020, doi: 10.1109/TNSE.2020.2995863.
- [11] Z. Yan, J. Ge, Y. Wu, L. Li, and T. Li, "Automatic virtual network embedding: A deep reinforcement learning approach with graph convolutional networks," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 6, pp. 1040–1057, Jun. 2020.
- [12] H. K. Thakkar, C. K. Dehury, and P. K. Sahoo, "MUVINE: Multi-stage virtual network embedding in cloud data centers using reinforcement learning-based predictions," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 6, pp. 1058–1074, Jun. 2020.
- [13] Z. Li, Z. Lu, S. Deng, and X. Gao, "A self-adaptive virtual network embedding algorithm based on software-defined networks," *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 1, pp. 362–373, Mar. 2019.
- [14] M. Demirci and M. Ammar, "Design and analysis of techniques for mapping virtual networks to software-defined network substrates," *Comput. Commun.*, vol. 45, pp. 1–10, Jun. 2014.
- [15] M. K. Haghani, B. Bakhshi, and A. Capone, "Multi-objective embedding of software-defined virtual networks," *Comput. Commun.*, vol. 129, pp. 32–42, Sep. 2018.
- [16] P. Zhang, H. Yao, and Y. Liu, "Virtual network embedding based on computing, network, and storage resource constraints," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3298–3304, Oct. 2018.
- [17] M. Pham, D. B. Hoang, and Z. Chaczko, "Congestion-aware and energy-aware virtual network embedding," *IEEE/ACM Trans. Netw.*, vol. 28, no. 1, pp. 210–223, Feb. 2020.
- [18] X. Cong, K. Shuang, and L. Zi, "Energy optimized virtual network embedding with location constraint in the enterprise network," *IEEE Access*, vol. 8, pp. 56170–56180, 2020.
- [19] A. Fischer, J. F. Botero, M. T. Beck, H. de Meer, and X. Hesselbach, "Virtual network embedding: A survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 4, pp. 1888–1906, 4th Quart., 2013.
- [20] 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, Mar. 2008.
- [21] M. R. Rahman and R. Boutaba, "SVNE: Survivable virtual network embedding algorithms for network virtualization," *IEEE Trans. Netw. Service Manage.*, vol. 10, no. 2, pp. 105–118, Jun. 2013.
- [22] K. Zhu, Z. Yao, N. He, D. Li, and L. Zhang, "Toward full virtualization of the network topology," *IEEE Syst. J.*, vol. 13, no. 2, pp. 1640–1649, Jun. 2019.
- [23] Z. Yan, N. Wei, Q. Jin, and X. Zhou, "Latency-aware resource-efficient virtual network embedding in software defined networking," in *Proc. 28th Wireless Opt. Commun. Conf. (WOCC)*, May 2019, pp. 1–5.
- [24] H. Huang, S. Li, K. Han, Q. Sun, D. Hu, and Z. Zhu, "Embedding virtual software-defined networks over distributed hypervisors for vDC formulation," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.
- [25] E. Guler, D. Zheng, G. Luo, L. Tian, and X. Cao, "Embedding virtual multicast trees in software-defined networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.
- [26] S. S. W. Lee, K.-Y. Li, W.-K. Liu, C.-H. Chen, H.-J. Fang, and T.-S. Wong, "Embedding bandwidth-guaranteed network-based virtual Ethernet switches in SDN networks," *J. Lightw. Technol.*, vol. 35, no. 23, pp. 5041–5055, Dec. 1, 2017.
- [27] J. de Jesus Gil Herrera and J. F. B. Vega, "Network functions virtualization: A survey," *IEEE Latin Amer. Trans.*, vol. 14, no. 2, pp. 983–997, Feb. 2016.
- [28] P. Zhang, H. Yao, C. Qiu, and Y. Liu, "Virtual network embedding using node multiple metrics based on simplified ELECTRE method," *IEEE Access*, vol. 6, pp. 37314–37327, 2018.
- [29] 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 Comput. Commun. Rev.*, vol. 41, no. 2, pp. 39–47, 2011.
- [30] N. M. M. K. Chowdhury, M. R. Rahman, and R. Boutaba, "Virtual network embedding with coordinated node and link mapping," in *Proc. IEEE INFOCOM*, Rio de Janeiro, Brazil, Apr. 2009, pp. 783–791.
- [31] E. Rodriguez, G. P. Alkmim, N. L. S. D. Fonseca, and D. M. Batista, "Energy-aware mapping and live migration of virtual networks," *IEEE Syst. J.*, vol. 11, no. 2, pp. 637–648, Jun. 2017.
- [32] L. Gong, Y. Wen, Z. Zhu, and T. Lee, "Toward profit-seeking virtual network embedding algorithm via global resource capacity," in *Proc. IEEE INFOCOM-IEEE Conf. Comput. Commun.*, Apr. 2014, pp. 1–9.
- [33] P. Zhang, H. Yao, and Y. Liu, "Virtual network embedding based on the degree and clustering coefficient information," *IEEE Access*, vol. 4, pp. 8572–8580, 2016.



SHEN LING (Member, IEEE) was born in 1987. He received the master's degree in communications engineering from Donghua University, Shanghai, China, in 2014. He is currently pursuing the Ph.D. degree with the Beijing University of Posts and Telecommunications (BUPT). His research interests include network virtualization and software defined networks.



WU MUQING (Member, IEEE) was born in 1963. He received the Ph.D. degree. He is currently a Professor with the Beijing University of Posts and Telecommunications (BUPT). His research interests include new network architecture, communication networks, high-speed network traffic control and performance analysis, and GPS locating and services. He is a Senior Member of the China Institute of Communications.



XIAOLAN HOU (Member, IEEE) was born in 1988. She received the master's degree in communications engineering from Yanshan University, Hebei, China, in 2016. She is currently pursuing the Ph.D. degree with the Beijing University of Posts and Telecommunications (BUPT). Her research interest includes controller placement for software defined networks.

• • •